National Library of Canada
Canadian Theses Service
Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada
Service des thèses canadiennes

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

NL-339 (r. 88/04) c

Canada

THE UNIVERSITY OF ALBERTA

ATOMIC SPECTROCHEMICAL MEASUREMENTS USING A FOURIER TRANSFORM SPECTROMETER

by

GREGORY BRIAN KING

A Thesis
Submitted To The Faculty Of Graduate Studies And Research
In Partial Fulfilment Of The Requirements For The Degree
Of Doctor Of Philosophy

The Department of Chemistry

Edmonton, Alberta
Spring, 1989

National Library
of Canada

Canadian Theses Service

Ottawa, Canada
K1A 0N4

Bibliothèque nationale
du Canada

Service des thèses canadiennes

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-52850-8

Canada

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: Gregory Brian King

TITLE OF THESIS: Atomic spectrochemical measurements using a Fourier transform spectrometer.

DEGREE FOR WHICH THESIS WAS PRESENTED: Ph.D.

YEAR THIS DEGREE GRANTED: 1989

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

....
(signed)

SCIEX,
#202, 55 Glen Cameron Road,
Thornhill, Ont.
L3T 1P2
(416) 881-4646

Date: Jan. 16 1989

Finagle's law:

The perversity of the universe tends to a maximum.

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled ATOMIC SPECTROCHEMICAL MEASUREMENTS USING A FOURIER TRANSFORM SPECTROMETER submitted by GREGORY BRIAN KING in partial fulfilment of the requirements for the degree of Doctor of Philosophy.

Gary Horlick
Dr. G. Horlick (supervisor)

Frederick F. Cantwell
Dr. F. F. Cantwell

John E. Bertie
Dr. J. E. Bertie

James A. Plambeck
Dr. J. A. Plambeck

I G Dalla Lana
Dr. I. G. Dalla Lana

Dr. W. G. Fateley (external examiner)

Date: Jan 16 1987

This thesis is dedicated to Steven Jobs,

for having the vision.

Abstract

The application of Fourier transform spectroscopy (FTS) to analytical atomic spectroscopy has been hampered by the phenomenon known as the multiplex disadvantage which increases the amount of background noise in the measured spectra. This problem is inherent in the technique and though it has been thoroughly studied from a theoretical point of view there has been little empirical study of the phenomenon.

The effects of experimental parameters, such as instrument resolution and interferogram sampling rate on both the signal and background noise will be presented in this thesis. The spectral distribution of the background noise is also shown. A theoretical model has been derived that accounts for both the noise behavior and spurious spectral features. The use of multiple spectral lines to construct calibration curves and calculate detection limits will be considered.

Some of the desirable features of FTS are shown by using the technique to acquire inductively coupled plasma (ICP) atomic emission spectra of some selected elements. Its high wavelength accuracy, simultaneous bandpass coverage, and high throughput make FTS particularly suited for compiling spectral data. The acquired data was then used to implement an automated elemental analysis system using a simple rule-based expert system.

In addition, advanced programming concepts and user interfaces will be presented from the point of view of how they can help the scientific researcher. Implementation of data manipulation and display routines for very large data sets will be described. A brief review of the various algorithms for calculating the Fourier transform is also included in this thesis.

Preface

Good software is an example of evolution in action, it is constantly evolving as bugs are eradicated and new features implemented. In addition the dedicated programmer is constantly fine tuning the code in order to make it smaller and faster. It is for this reason that I have avoided, wherever possible, including the actual source code for my routines and have instead chosen to use pseudo-code.

Actual source code for some of my routines only appear in the appendices. The code is fully functional and correct but since writing this thesis I have changed development systems from the original MDS format TML Pascal 2.00 to the new TML Pascal II running under MPW. This required many changes to the format of the code and the linking schema. The actual routines remain unchanged, therefore it was deemed unnecessary to change the appendices.

As a note in passing, the change in compilers resulted in a 2% reduction in code size of the SpectroPlot program.

This thesis was prepared entirely on an Apple Macintosh Plus personal computer using Microsoft Word, MacDraw II, Expressionist, and Trapeze and was printed out on an Apple LaserWriter Plus printer.

Acknowledgements

Scientific research today is very much a team effort and I wish to thank the other members of my team.

The two people without whom this work would not have been possible are Professor Gary Horlick, for his advice, financial support, and patience and Bruce Todd, builder of the world's best small uv-vis interferometer.

The rest of our research group, both past and present, contributed general abuse (none of it undeserved). However, one person does stand out (by seniority?) from the rest. Vassilli Karanassios introduced me to programming the Macintosh and I have spent much time discussing with him the future of analytical instrumentation.

Many thanks also to Professor Bertie for all his useful comments on the manuscript and my parents for their patience. Too numerous to mention individually are all the other members of the Chemistry department who made my tenure memorable.

I also wish to acknowledge Dash the Dorg, a truly unique creature who had a different perspective on life and the folks at Softwarehouse (West) Inc. from whom I bought $8000 worth of software and books and who always let me test fly the latest software.

TABLE OF CONTENTS

CHAPTER PAGE

1. Introduction 1
   - 1-1. The spectrometer 2
   - 1-2. The data system 9
   - 1-3. Programming the Macintosh 12
   - 1-4. The electronic spectrograph: Star Trek spectroscopy 15
   - 1-5. Overview of this thesis 17
2. Algorithms for Fourier Transform Spectroscopy 20
   - 2-1. The Cooley–Tukey FFT algorithm 23
   - 2-2. The Prime factor algorithm 24
   - 2-3. The Hartley transform 25
   - 2-4. Optimizing the Cooley–Tukey algorithm 27
   - 2-5. Conclusions 41
3. Implementing new concepts in user interfaces 42
   - 3-1. Design paradigms 44
   - 3-2. Interacting with data 49
   - 3-3. Handling large data sets 57
   - 3-4. Flexibility 61
   - 3-5. Programming paradigm 67
   - 3-6. Conclusions 68
4. Qualitative atomic spectrochemical analysis 70
   - 4-1. Aliasing 70
   - 4-2. Wavelength calibration 75

4-3. Resolution 83

4-4. Spurious spectral lines 90

4-5. Conclusions 103

5. Quantitative atomic spectrochemical performance 104

5-1. Measuring noise 105

5-2. The multiplex disadvantage 110

5-3. Detection limits 123

5-4. Dynamic range 129

5-5. Conclusions 135

6. A spectral atlas of ICP atomic emission lines for selected elements 137

6-1. Experimental parameters 138

6-2. Choosing the number of scans 139

6-3. Line assignment 141

6-4. Unassigned lines 142

6-5. Results 142

6-6. Conclusions 148

7. Automatic identification of spectral lines in ICP-FTS spectra 154

7-1. Defining and finding peaks in atomic emission spectra 155

7-2. Assigning peaks in atomic emission spectra 161

7-3. A rule-based expert system for identifying elements from atomic emission spectra 167

7-4. The determination of major elements in brass, steel, and stainless steel 170

7-5. Conclusions 186

8. Multiline detection limits 190

8-1. Theory 191

8-2. Fe ICP detection limit 194
8-3. Conclusions 198
9. Summary and overall conclusions 202
9-1. Improvements in ICP performance 204
9-2. Software improvements 206
9-3. Future directions in analytical instrumentation 206

Bibliography 214

APPENDIX PAGE

A. National Instruments MIO–16 initialization program 223
B. National Instruments MIO–16 interface code 224
C. Data acquisition desk accessory for the Macintosh 236
D. Assembly language FFT listing for Macintosh II 241
E. ICP lines for 21 elements 245
F. ICP spectra of 19 elements 274

LIST OF TABLES

TABLE PAGE

1. Comparison of high level language and assembly language implementations of FFT algorithm. IBM times are for a 4096 point transform while the Macintosh times are for a 16384 point transform. Math coprocessors and non-optimizing compilers were used on both machines. MS Fortran 3.13 and TML Pascal 2.5 used on IBM and Macintosh respectively. 36

2. Aliasing regions in both wavenumbers and nanometers for sample clocks of 1, 4, and 8 times the basic HeNe clock frequency. 72

3. Observed wavelength values for the Mg quintuplet when the wavelength axis is calibrated using the reference value for the HeNe wavelength. 78

4. Observed wavelength values for the Mg quintuplet when the wavelength axis is calibrated using a corrected value for the HeNe wavelength calculated from NBS Fe reference lines 80

5. Lines used to calibrate the wavelength region between 190 nm and 200 nm 84

6. Calculated wavelength resolutions at various wavelengths for a 1.0 cm^{-1} wavenumber resolution 86

7. Measured and calculated modulation frequencies from HCL modulation experiment 98

8. List of elements and the exact concentrations used in acquiring their spectra 140

9. Overall summary of atomic lines database 146

10. Sn ICP emission lines 147

11. Lines observed in aliased and de-aliased ICP spectra of Pb 151

12. Certified composition of NBS stainless steel SRM 121d 171

13. Certified composition of NBS low alloy steel SRM 362 172

14. Certified composition of NBS Naval brass B SRM 1107 173

15. Software assigned lines for SRM 121d. 179

16. Result of software analysis of SRM 121d spectra. 184

17. Fe lines used for multiline calibration. 197

LIST OF FIGURES

FIGURE PAGE

1. Overall block diagram of instrument 5

2. Pseudo-code for a direct implementation of the Cooley–Tukey radix-2 decimation in time algorithm. All non-integer operations are complex. 28

3. A more efficient implementation of the Cooley–Tukey radix-2 decimation in time algorithm obtained by reversing the two inner loops. 30

4. FFT algorithm optimized for execution on digital computers 32

5. Pascal source code for Macintosh FFT 33

6. Pascal definition for the xPtr type. 35

7. Pascal source code for bit reversal routine. 39

8. The complete Macintosh FFT routine. 40

9. Data windows from SpectroPlot showing the various cursors (a) and a spectrum with a selected region (b). 52

10. An example of a typical dialog (a) and an error alert (b). 54

11. Some more examples of dialogs 55

12. A typical SpectroPlot menu 56

13. Sample Pascal source code showing direct access of data using pointers and pointer math. 59

14. Pascal header for a SpectroPlot external procedure (a) and definition of parameter block (b) 64

15. An example of SpectroPlot output that has been modified using a graphics program. 66

16. Aliased Mg HCL spectrum showing the wavelength regions when an R166 PMT is used as a detector (a) and a close-up of the quintuplet at 277.983 nm (b). 74

17. Aliased Fe ICP spectrum (a) showing the wavelength regions corresponding to the de-aliased spectrum (b)........ 76

18. Aliased Fe ICP spectrum (a) and a portion of the Fe spectrum showing the NBS wavelengths used to calibrate the wavelength axis (b)........ 79

19. Actual spectral data and interpolated lineshapes for Mg 279 (a) and 280 (b) nm emission lines........ 82

20. Air-vacuum wavelength calibration curve........ 85

21. Hollow cathode emission of the Zn 213 nm line, 64K interferogram........ 88

22. An aliased 0.5 nm portion of an ICP spectrum of Fe calculated from a 64K interferogram. 1000 mg/L Fe........ 89

23. Physical set-up for modulation experiment........ 94

24. Unmodulated (a) and 146 Hz modulated (b) Mg hollow cathode lamp spectra........ 95

25. Plot of sideband frequency as a function of separation showing both calculated and observed frequencies........ 97

26. Mg ICP spectrum (a) and close-up of region around primary emission lines showing modulation sidebands(b)........ 100

27. Reduction of modulation sideband intensity by signal averaging showing both observed and predicted values........ 101

28. Scale expanded view of a Mg ICP spectrum showing the plasma modulation bands on either side of the parent peaks (a) and a close-up view of the left bands (b)........ 102

29. Outline of procedure for calculating standard deviation and signal-to-noise ratio spectra........ 107

30. Intensity (a), standard deviation (b), and signal-to-noise ratio (c) spectra of Mg ICP emission. Calculated from 16K aliased interferograms........ 108

31. Close-up of Mg spectra showing intensity (a) and standard deviation (b) spectra. From 4K 10 mg/L Mg ICP interograms sampled with laser clock (aliased)........ 113

32. Close-up of Mg spectra showing intensity (a) and standard deviation (b) spectra. From 16K 10 mg/L Mg ICP interograms sampled with laser clock (aliased)........ 114

33. Closer views of Mg standard deviation spectra calculated from 16K interferograms........ 115

34. Plot of background noise as a function of Mg concentration in an aliased spectrum........ 117

35. Scale expanded, de-aliased ICP spectrum of Mg showing distribution of baseline noise........ 118

36. Region of de-aliased spectrum (a) corresponding to an aliased spectrum (b)........ 120

37. Plot of baseline noise as a function of Mg concentration from a de-aliased spectrum........ 121

38. Aliased ICP spectra of Zn (a) and Fe (b) showing the relative distribution of their lines........ 122

39. The effect of 1000 mg/L Fe on the aliased spectrum of 1 mg/L Zn. Spectra of 1 mg/L Zn (a), 1000 mg/L Fe (b), and 1 mg/L Zn + 1000 mg/L Fe (c)........ 124

40. The effect of accumulating interferograms on dynamic range (a), noise (b), and signal intensity (c)........ 131

41. The effect of interferogram length on dynamic range (a), signal intensity (b), and baseline noise (c)........ 133

42. 64K interferogram of Mg ICP emission........ 134

43. The ultraviolet ICP spectrum of Ta (a) and an ~2 nm close-up of the region containing the most intense emission lines (b)........ 143

44. The overall view of the spectral line database (a) and a sample line data record from Reflex Plus (b)........ 145

45. A sample screen showing the use of an online database to identify a Si line in a W spectrum........ 149

46. Scale expanded, de-aliased ICP spectrum of Pb showing distribution of baseline noise........ 153

47. Procedure for estimating the baseline mean and standard deviation of a spectrum........ 157

48. Two spectral simulations using uniformly distributed noise. Peaks that are 3, 9, 4, and 5 times the standard deviation of the baseline are located at points 25, 40, 50, and 70, respectively........ 159

49. Two spectral simulations using Gaussian distributed noise. Peaks that are 3, 9, 4, and 5 times the standard deviation of the baseline are located at points 25, 40, 50, and 70, respectively 160

50. A frequency histogram for a baseline region in a spectrum of 10 mg/L Mg. The values are scaled to show the relative distance from the mean baseline value in units of standard deviation. 162

51. Typical results from a spectral search of a aliased Zn ICP spectrum with tentative matches from a database of literature values. 165

52. Peaks from a Zn ICP spectrum showing minor spectral features in addition to the three Zn lines. 166

53. A rule for determining the presence of Mg. 169

54. The ICP ultraviolet spectrum of NBS stainless steel SRM 121d. 175

55. The ICP ultraviolet spectrum of NBS low alloy steel SRM 362. 176

56. The ICP ultraviolet spectrum of NBS brass SRM 1107 177

57. Expanded views of the regions ~1 nm on either side of the three principle Ni lines in the spectrum of SRM 121d. 180

58. Expanded views of the regions ~1 nm on either side of the three principle Mn lines in the spectrum of SRM 121d. 181

59. Expanded views of the regions ~1 nm on either side of the three principle Cr lines in the spectrum of SRM 121d 182

60. Output from expert system 183

61. Scale expanded views of the spectrum of SRM 121d where principle lines of Cu (a), Co (b), and Si (c) would be found. 185

62. Graphical depiction of the detection limit definition. 192

63. Simulated spectrum with equal intensity lines 195

64. Simulated spectrum with only 2 major lines. 196

65. Multiline calibration curve for Fe. 199

66. Plot of theoretical multiline detection limit gain (dI/dI_N) vs. number of lines (N) for the 15 most intense Fe lines in the ultraviolet. 200

67. Conventional configuration for instrument automation. 208

68. Distributed instrumentation with remote solution handling, "chemical workstations", and computer workstations on a local area network. 209

69. Block diagram of a single intelligent instrument unit. 212

Chapter 1

Introduction.

An ideal spectrometer would have as its virtues, a wide spectral band-pass, simultaneous coverage of its entire band-pass, high throughput, rapid acquisition of spectra, absolute intensity and wavelength accuracy, wide dynamic range, low noise, and computer compatibility. No spectrometer in existence today can meet all these requirements but we believe that Fourier transform spectroscopy is one way to meet them.

Fourier transform spectrometers already meet most of the requirements for the ideal spectrometer, the challenge is to improve their performance in those areas where it is still deficient. The most notable difficulty is the multiplex disadvantage in the visible and ultraviolet regions of the electromagnetic spectrum which is inherent in the method. Other problems such as tighter mechanical tolerances (which affect resolution), velocity control requirements (spurious peaks and background noise), and handling very high data bandwidths are purely technical and largely solved.

There currently exist several Fourier transform spectrometers designed specifically for operation in the ultraviolet–visible (uv-vis) spectral region [1 – 4]. They are large high resolution instruments more suitable for astronomical [1, 2] and fundamental work [3, 4] than for routine analysis. Bomem and Bruker offer smaller commercial instruments but they are modified versions of existing infrared instruments. The goal of our research is the design of a compact uv-vis Fourier transform spectrometer suitable for routine operation in an analytical laboratory.

Previous students in our laboratory have built and studied uv–vis Fourier transform spectrometers for over a decade while our supervisor, Gary Horlick, has two decades of experience. Horlick was an early proponent of Fourier transform spectroscopy (FTS) as a measurement system for atomic spectrochemical analysis [5 – 9]. Yuen built the first generation of the instrument used in our laboratory [12] and used it to study flame atomic emission [10, 11]. The optical emission from the inductively coupled plasma (ICP) was studied using FTS by Hall [13]. Ng studied data processing [14] and correlation methods [19, 20]. Stubley modified the instrument [16] and used it to assess both near ir [15] and uv-vis [17] performance of ICP–FTS. In addition she also studied some effects of optical pre-filtering the input radiation [18]. The first comprehensive noise study of ICP–FTS was undertaken by Marra [21].

Much of this early work was hampered by low resolution, aliasing, insufficient data bandwidth, and inadequate computational capabilities. With recent advances in computer and digital signal processing technology along with an innovative new design for controlling the motion of the moving mirror we believe we have finally achieved our goal of developing a usable compact, low cost uv-vis Fourier transform spectrometer.

1-1. The spectrometer

Our current Fourier transform spectrometer (FTS) was designed and built in our laboratory by Bruce Todd [22]. It is based on a standard configuration Michelson interferometer with the fixed and moving mirrors arranged 90° to each other and a 45° beamsplitter between them. The moving mirror is suspended from the sleeve of a rectangular air bearing to eliminate torsional and extensional deflections. The basic interferometer cube design is the same as previously used in our laboratory [12, 16].

One of the difficulties in building interferometers for use in the ultraviolet is maintaining the required mechanical and alignment tolerances. This is because such tolerances are proportional to the operational wavelengths, which are in the micrometer range for infrared but are tenths of a micrometer for the ultraviolet.

In addition to mechanical tolerances, the velocity control of the moving mirror must also have tighter tolerances. Our mirror drive has a constant velocity because velocity variations can lead to increased noise and spurious peaks in the calculated spectra. These can occur in spite of using a digitization clock (which triggers the ADC) derived from a laser interferogram. While this ensures proper sampling at precise increments of optical retardation, errors can still occur due to an apparent modulation of the source and phase delays in the electronics.

The required interferogram sampling rates for a given mirror velocity are also much higher in the uv than in the ir (if aliasing is to be avoided) because the frequency of the interferogram waveform is inversely proportional to the wavelength. This places greater demands on the data acquisition system. Another consequence of the higher sample rates is the greater amount of data required for a given resolution which makes further demands on the data system.

Previous generations of our interferometer have been limited by their low resolution (8 cm^{-1}) and lack of precise velocity control. The current system has a 2 cm maximum optical retardation (0.5 cm^{-1} nominal resolution) and controls the velocity of the moving mirror to better than one part in a thousand (10,000 Hz ± 5 Hz laser frequency).

A basic overview block diagram of the instrument is shown in Figure 1. The basic interferometer has three opto-electronic channels; in addition to the illustrated sample channel there are white light and laser reference channels. The sources for the two auxiliary channels are a tungsten incandescent light bulb and a 632.8 nm HeNe laser, respectively. The white light channel is used to establish the zero path difference (ZPD) position of the moving mirror while the laser reference is used to control the mirror velocity and data acquisition sampling rates.

The filters are 48 dB/octave analog units for limiting the bandpass of the interferogram. They remove high frequency noise and low frequency modulation components as well as leakage from the auxiliary channels into the analog sample interferogram signal. These filters are a necessary evil since they can also introduce phase shifts into the analog data.

The interferometer control electronics are responsible for maintaining a constant mirror velocity and controlling data acquisition by the computer and contain a mix of analog and digital circuitry. Two active feedback circuits are used to control the mirror velocity. A moving magnet/coil system is used for coarse velocity control while the frequency of the laser interferogram is used for fine control. Only one is used at any given moment and internal logic is used to switch between the feedback circuits.

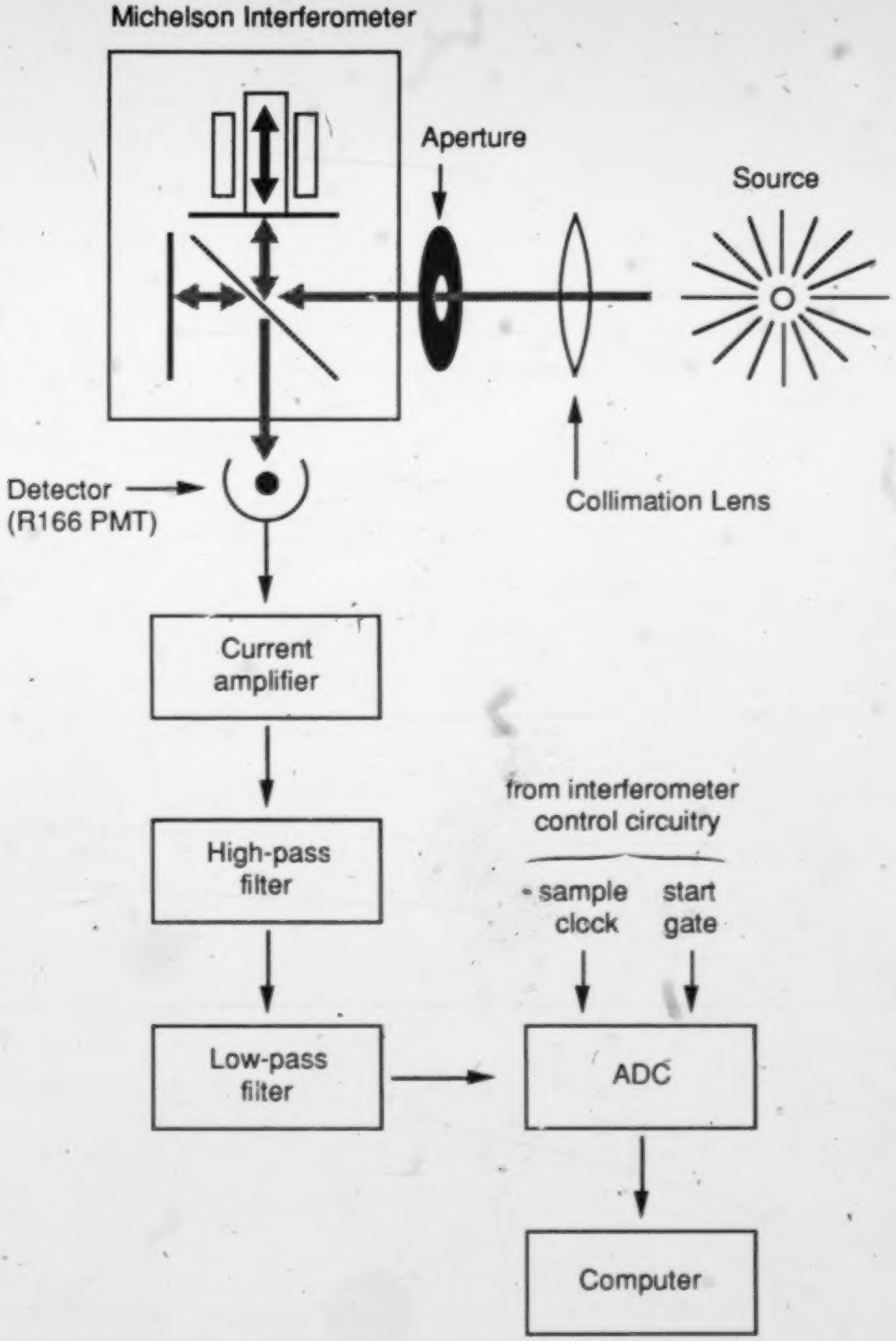

Michelson Interferometer
Aperture
Source
Detector
(R166 PMT)
Collimation Lens
Current amplifier
High-pass filter
Low-pass filter
from interferometer control circuitry
sample clock
start gate
ADC
Computer

Figure 1. Overview block diagram of instrument.

The dual feedback circuit is an essential feature of our instrument and allows control to be maintained even at the mirror turnaround points. The coarse feedback circuit allows control to be maintained throughout the turnaround procedure while the fine feedback is sufficiently sensitive to maintain the high degree of velocity control required during data acquisition.

The moving mirror drive is of a dual voice coil design and is set up in such a way that when the drive is at the correct velocity no current flows through the coils. In this manner, the voltage across the coils is always an amplified version of the error voltage generated by the velocity control circuitry.

The data acquisition portion of the control electronics generates a sample clock for the ADC and a start pulse at the beginning of each scan. The sample clock is derived from the laser interferogram with a base frequency identical to that of the laser interferogram. A clock at 4 times the base frequency is generated with a phase locked loop (PLL) frequency synthesis circuit and an 8 times clock is then generated from this using a pair of monostables firing on opposite zero crossings.

The start of data acquisition is determined by continuous counting of clock pulses. The first mirror turnaround point is set for N data points by subtracting N/2 from the number of clock cycles between the rear limit switch and ZPD. The second turnaround point is set by adding N to the start of the forward scan. The electronics continues to count clock pulses from when turnaround starts to when the velocity reaches zero, this value is used to determine when the drive returns to the start of turnaround position going in the opposite direction. Each time after the drive has successfully turned around a start pulse is sent to the computer.

Once the turnaround points have been set the drive will oscillate between them, the absolute position of the drive with respect to the rear limit switch is always known. This is possible because the drive is always under velocity control, even at turnaround. Currently the number of data points in a scan is set independently on both the spectrometer and the computer though it is possible in future instruments for the computer to send the value to the spectrometer.

Double sided interferograms are produced by the instrument, an N point interferogram samples N/2 points on either side of ZPD. By calculating amplitude spectra from such interferograms, the question of phase correction is eliminated. This also provides the equivalent of one level of zero filling to the final spectrum for better digital resolution in the wavelength axis [23]. The spectra calculated from our N point double sided interferograms are equivalent to those calculated from N/2 point single sided interferograms that have been zero filled to N points. Note that this does not improve the nominal resolution of the instrument which is strictly determined by the maximum optical retardation.

At the basic HeNe clock rate (10 kHz), 65,536 points are acquired when the interferometer is scanned over the full ±2 cm of optical retardation of the instrument. The drive traverses the entire range in 6.5 sec which translates to a mirror velocity of ~0.3 cm/sec. Depending upon the actual sampling rate used, 0.5 cm^{-1} resolution interferograms can contain from 65,536 to 524,288 data points.

A major difference between our instrument and other currently existing ones is the velocity of the moving mirror. Because the interferogram sampling rate is proportional to the mirror velocity it serves as a convenient indicator. The peak sampling rate for uv-vis work is nominally defined as the minimum clock rate which satisfies the Nyquist requirement for wavelengths down to 158 nm. This translates to a frequency eight times greater than that of a 632 nm HeNe laser interferogram.

Both the Kitt Peak [1] and the Thorne [4] interferometers have slow drives and are only capable of a maximum rate of 5 kHz. Because of this, high resolution scans take a long time and hence, require a very stable source. The Los Alamos instrument [3] can achieve 50 kHz rates and our instrument has been used at 80 kHz. Because our drive is fast it is more forgiving of source drift than other existing instruments.

Another important feature of our instrument is a duty cycle approaching 100%. Data is acquired on both the forward and back scans. Only for an approximately 150 msec period for turnaround is the instrument inactive. This speed of acquisition of multiple interferograms adds greater immunity to source drift.

Quartz optics are used for coverage down into the uv and the detector can either be a 1P28 or an R166 photomultiplier tube, though all the work described in this thesis was done with the R166 tube. The beamsplitter assembly is composed of a 100 Å layer of Al on a Suprasil 1 substrate with a Suprasil 1 compensator plate. This beamsplitter has its maximum efficiency in the ultraviolet. Both mirrors are front surface aluminum with a MgF_2 overcoat.

The entire instrument is open to the room air which precludes wavelength coverage below 190 nm. Stubley [18] had purged the optical path of a previous generation of this instrument with argon to extend the uv response to 180 nm. An aperture is necessary in the design to limit off-axis rays which can broaden the instrument line width. Currently an 8 mm aperture is used.

1-2. The data system

The current data system used with our FTS system is an Apple Macintosh II workstation. Previous generations of the instrument have used DEC PDP-11/10 and IBM PC-XT computers. The instrument was designed to be computer independent so that whenever advances in computer technology occur they can be taken advantage of by simply substituting the newer machine for the current computer.

The Macintosh II has a full 32 bit architecture that uses a Motorola 68020 central processor and a Motorola 68881 floating point processor with a 16 MHz system clock. Currently up to 8 MBytes of RAM can be installed on the motherboard with only 2 MBytes actually installed in our system. When 16 MBit memory chips become available the limit will be 128 MBytes. Additional memory can also be installed on add-on boards and a Motorola 68851 paged memory management unit is optional. Six NuBus slots are available on the motherboard for third party hardware.

The video display is a 13" Apple colour monitor connected to an Apple video card with the 8 bit memory option installed. The monitor is 640 × 400 pixels in size with a resolution of 75 ppi. In this configuration, it is possible to display 256 simultaneous colours from a palette of 16M. Third party monitors are available up to 36" in size with more colours and greater numbers of pixels. The operating system also has built-in support for multiple monitors that is user configurable.

Mass storage consists of two 40 MByte hard disks, an internal Apple drive and an external ByteBox 40. The external unit is used for sharing data with other Macintosh computers in the lab. The system unit also contains an internal 800K 3.5" floppy drive.

Output is onto an Apple LaserWriter Plus laser printer. Other optional output devices include pen plotters, phototypesetters, and high resolution thermal colour printers. The most desirable output devices are those that support the Postscript page description language. Such devices are expensive since they contain as much, if not more, computing power than the host computer.

Data is acquired from the instrument through a National Instruments MIO–16H multifunction card. This card features a 16 channel, 100 kHz, 12 bit ADC, a 12 DAC, 8 digital I/O lines, and 5 16 bit timer/counters. An optional DMA card is available for maximum throughput but 100 kHz data rates can be achieved in short bursts without it.

One peculiarity of the card's design is that it doesn't properly reset itself when powered-up. This causes it to severely load the clock output of our instrument. The solution to this was a piece of initialization code that is executed whenever the Macintosh restarts. The source code for this INIT resource is given in Appendix A.

It was decided to write our own low level software interface to the data acquisition card instead of using the drivers supplied by National Instruments because of the need to achieve the maximum performance possible from the hardware. Appendix B contains the source for our interface to the card and serves as an example of how the card can be accessed from software.

The main feature of the Macintosh NuBus architecture is that it also specifies a software interface to peripheral cards in addition to the hardware. This allows application software to query each slot looking for devices that it can use and makes manually specifying slot addresses a thing of the past. An example of such a self-configuring application is the desk accessory listed in Appendix C. This desk accessory allows data to be acquired from within any program and is especially useful with spreadsheets.

When the desk accessory is opened it searches the slots for the first ADC card it can find and uses it as the default device. The user can also specify another slot if more than one ADC is installed. The software checks whether the slot entered contains a valid ADC card before attempting to acquire data. The combination of intelligent hardware and the proper software eliminates errors due to incorrect hardware configurations.

We chose the Macintosh II because of its advanced user interface and its computational power. It is rated at 2 MIP's and performs double precision floating point operations at better than two-thirds the speed of an unloaded (single user) VAX-11/780 [24]. Another consideration was the linear memory model of the Motorola chip in contrast to the segmented memory architecture of Intel chips used in MS-DOS and OS/2 machines.

Recently [25] the Los Alamos FTS Facility has chosen to replace their VAX class computers with Macintosh II's for much the same reasons I have described. In addition, they have discovered the productivity benefits of real time interactive data processing afforded by dedicated workstations.

1-3. Programming the Macintosh

A primary reason behind using an industry standard computer such as the Macintosh is the availability of commercial software packages. However, because of the special software requirements necessary to acquire data from our instrument and to handle large amounts of spectral information some custom programming was required.

Writing programs for advanced event driven user interface management systems such as the Macintosh OS, Microsoft Windows, or OS/2 Presentation Manager is a non-trivial task. The use of events and messages requires major shifts in programming models and adherence to guidelines is essential. Each of these environments supply several hundred subroutines for managing the user interface. This makes the programmer's life easier in that there is no need to "re-invent the wheel", but they must all be mastered before an application can be written, a truly formidable learning curve. Typically six months to a year is required to become proficient in any of these environments assuming prior programming experience in conventional systems.

The extensive use of structured data demand that structured languages such as C, Pascal, or Modula-2 be used. More traditional languages such as Fortran or Basic are clearly unsuitable. Even debugging is an adventure in itself with non-linear code execution, dynamically allocated relocatable memory blocks, inter-application compatibility, and multiple tasks.

So it is clear that good software tools and information are essential to mastering programming the Macintosh. Software tools include language compilers, text editors, debuggers, and resource editors. Each of these tools can be independent stand-alone components or they can be combined into integrated programming environments. Integrated programming environments can greatly speed the development cycle.

Good sources of information are the key to successful programming of the Macintosh. The Macintosh ROM's contain 718 subroutines, most of which are required to build a fully functional program. In learning to use these routines the Macintosh programmer must read and absorb thousands of pages of documentation. At this time there are several good texts on Macintosh programming which serve to supplement the Apple manuals.

Software information tools are a hybrid of program and software manuals, being an online database that allows instant access to programming information. Because of the complexity of the Macintosh programming environment, such online databases are most useful.

The two most popular Macintosh development systems, Lightspeed C and Pascal, are integrated environments that combine text editor, compiler, linker, and debugger all into one program. Both are powerful systems used for developing commercial applications, yet are friendly enough for novice users. Two other integrated systems are Borland's Turbo Pascal for beginners and Apple's Macintosh Programmer's Workshop for professional programmers.

Other products available are Consulair's Mac C, which was the first development system for the Macintosh, and TML Pascal. The Consulair system has fallen into disuse while the TML compiler is migrating from the Consulair object format to MPW. Basic and Fortran compilers are available but are not recommended for programming the Macintosh user interface.

The absolute authority on the routines contained within the Macintosh ROM is "Inside Macintosh" [26] from Apple. This set of volumes, currently five, contains descriptions of every routine but is meant to be a reference work for experienced programmers. Beginners are better off starting with "Macintosh Revealed" [27] which explains how to use the basic ROM routines to create an application that conforms to Apple guidelines.

For advanced programmers Knaster has two books [28, 29] describing more sophisticated techniques. Much of the existing documentation uses Pascal to illustrate various examples, this is because the Macintosh ROM routines were designed to be called from Pascal. C programmers are referred to Takatsuka *et al.* [30] and Mathews and Friedland [31] for the C calling conventions.

Those who wish to learn assembly language for the Mac should read Dan Weston's "The Complete Book of Macintosh Assembly Language Programming" [32] vol. 1. Unfortunately, the Consulair MDS assembler is described but much can be applied to the newer Apple MPW assembler. Volume 2 of this set contains several advanced example programs that would also be useful for programmers in high level languages.

Also useful are the "Apple Numerics Manual" [33] which describes in detail the floating point routines and "Designing Cards and Drivers for

Macintosh II and Macintosh SE" [34] which describes some aspects of the peripheral card interface. Unfortunately, the second reference doesn't quite contain enough information on using Macintosh II slot resources.

Other sources of information are MacTutor and Apple Macintosh Technical Notes. MacTutor is the only Macintosh programming journal in wide circulation and features many articles on programming some of the more esoteric Macintosh features. The Apple Technical Notes are distributed to Apple Certified Developers and contain bug reports, programming tidbits, workarounds, and other pieces of information useful to programmers.

A useful source of information and software tools is the Apple Programmers and Developers Association based in Renton WA. This is an independent association that is authorized by Apple to distribute official Apple information and software development tools. It is in effect a one stop shop for Apple programmers.

It has been facetiously said that the best way to write a Macintosh program is to hire someone who already knows how. In all.seriousness learning to program the Macintosh is a major undertaking requiring a lot of time, patience, skill, knowledge, and good taste in designing graphics. It also doesn't hurt to know someone who's already done it so that you have someone to turn to when you run into brick walls.

1-4. The electronic spectrograph: Star Trek spectroscopy

As it stands, Fourier transform spectroscopy is the electronic equivalent of the photographic plate spectrograph since it records an entire spectrum simultaneously and archives it in a manner such that it can be retrieved for future study. The major drawback of FTS and spectrographs is the large amount of information that can be recorded over a relatively short period of time, this is why archiving for future study is important.

Normal analytical procedure calls for determining a number of sought for species and all measurements are directed towards these species. If additional species are desired, then new measurements directed towards these species must be performed. In FTS everything is measured at once and when additional species are sought for, only the data needs to be re-interpreted, no new measurements are required. If the data has been archived, then other species can be studied in the future.

Because FTS stores its data in a machine readable form it is natural to consider combining it with computerized data reduction, processing, and interpretation. The automated handling of data is essential because of the large quantity of data produced by FTS and requires powerful computing hardware and more sophisticated software.

It is the combination of FTS with powerful software tools and fast computers that can make chemical analyses appear as trivial as depicted in science fiction. Currently an analytical chemist asks "Is such-and-such a species present in this sample?" This process is repeated for each species of interest and it is possible for an important species to be overlooked. The question that the chemist should be asking is "What is in this sample?", but it is normally impractical to ask this because of the time required to perform the measurements and interpret the data for all possible species.

Our goal is to design a software system that will take atomic emission data acquired from a Fourier transform spectrometer and automatically determine all detectable elements present in a sample. This in effect answers the question of what is in a sample without any *a priori* knowledge of the sample. Such a system must be able find all the spectral lines in the data, identify them, and resolve possible spectral interferences.

In atomic spectroscopy the wavelengths of emission lines are invariant (given the absence of strong magnetic fields and reasonable velocities below the speed of light) and this allows wavelength to be used to conclusively identify emitting species. This differs from the situation in molecular spectroscopy where chemical factors can shift emission wavelengths. When combined with the absolute wavelength accuracy of FTS it means that spectral lines can be assigned with a high degree of confidence.

Once all the lines have been tentatively assigned, which is done by simple matching with a database of known lines, chemical knowledge must be applied to the data to determine which emitting species are actually present. At this stage it is possible for a single spectral line to have more than one element assigned to it, due to the possibility of aliasing and/or spectral overlaps. An expert system is required to resolve the results of the database search and to extract the desired information from the data.

1-5. Overview of this thesis

The application of FTS to atomic spectrochemical analysis has been hampered by a degradation in performance relative to conventional grating instruments. Other than theoretical treatments there has been little systematic study of the mechanisms of degradation. In this thesis I will be characterizing and discussing the analytical performance of an FTS instrument built by B. Todd for use specifically in the uv-vis. I will also demonstrate how FTS can be used to perform automated elemental analysis.

The inductively coupled plasma was used in the majority of our studies since it is the primary atomic emission source used in our laboratory. The choice was mainly for convenience and does not preclude the use of other emission sources with FTS. The performance characteristics of the ICP-FTS system, with emphasis on its use as an electronic spectrograph, will be evaluated and discussed in this thesis.

A brief review of the various algorithms available to calculate the Fourier transform will be presented first. Since the Cooley-Tukey algorithm was formulated in the mid 1960's, other mathematicians have derived alternative algorithms which they claim will perform the transform faster. Details of an implementation of the Fourier transform algorithm optimized for execution on a digital computer will be presented.

Advanced programming concepts and user interfaces that facilitate productivity will also be discussed. This will include a detailed discussion of user interface requirements from the point of view of a researcher. Implementation of data manipulation and display routines for very large data sets will also be described with the emphasis on speeding operations so that they appear more interactive.

Some performance characteristics of ICP-FTS, as pertaining to its use as an electronic spectrograph, will be presented and discussed. Particular attention will paid to questions of aliasing, wavelength calibration and accuracy, spectral resolution, and spurious lines.

Some of the noise characteristics of ICP–FTS will also be presented, as they affect the quantitative performance of the technique. How noise is measured in a spectrum will be considered along with the effect of the multiplex disadvantage. The effect of instrument parameters such as length and number co-added of interferograms on signal intensity, background noise, and spectral dynamic range will be presented and discussed.

The capabilities of ICP–FTS will be demonstrated by using the technique to compile a spectral atlas of the ultraviolet emission of 22 elements. Lines from these spectra have been tabulated and entered into a machine database. The spectra are also available in machine accessible form.

Data acquired from the ICP–FTS system was then used to implement an automated elemental analysis system. The line detection and assignment algorithms and a simple rule based expert system will be described and their performance tested using some NBS standard reference materials. The software is very simple and was designed to prove that automated elemental analysis is possible using ICP–FTS.

Finally, the use of multiple analytical lines is proposed as a means of reducing the effects of the multiplex disadvantage on qualitative measurements. A calibration curve and detection limit will be calculated from the sum of several Fe lines.

Chapter 2

Algorithms for Fourier transform spectroscopy.

The concept of Fourier transform spectroscopy has been known since the 1890's [35, 36] but was not practical to implement until the invention of the digital computer. Even then the technique was limited by the amount of time required to compute the discrete Fourier transform (DFT), shown in Equation 2-1 for a sequence of arbitrary length N, which is required to recover the spectrum from the measured interferogram. The calculation is computationally intensive because the DFT involves multiplying an N element vector with an N × N matrix. The time required to calculate a transform of length N is proportional to N^2.

$$\overline{X}_k = \sum_{m=0}^{N-1} x_m W^{mk} \qquad \text{where} \quad k = 0, \ldots, N-1$$

$$W^{mk} = e^{-j2\pi mk/N}, \; j = \sqrt{-1} \qquad \text{Equation 2-1}$$

Because the time required to calculate a DFT increases geometrically with the size of the data set, it is not very practical to perform Fourier transform spectroscopy at very high resolutions and large spectral ranges using the DFT algorithm. This is not to say that it is impossible, it just requires a lot of computer power and patience. The DFT algorithm falls within a class of problems whose solutions have a polynomial time dependency. These are more tractable than algorithms with an exponential time dependency [37] which generally have no practical value.

Then in 1965 Cooley and Tukey [38] derived the fast Fourier transform (FFT) algorithm. The FFT algorithm recursively factors the data to reduce the number of operations required to calculate the transform. The general equations for the decimation in time radix–2 algorithm, which define the computational kernel, are:

$$\overline{X}_k = \sum_{m=0}^{N/2-1} x_{2m} W^{2mk} + W^k \sum_{m=0}^{N/2-1} x_{2m+1} W^{2mk} \qquad \text{and}$$

$$\overline{X}_{k+N/2} = \sum_{m=0}^{N/2-1} x_{2m} W^{2mk} - W^k \sum_{m=0}^{N/2-1} x_{2m+1} W^{2mk} \qquad \text{Equation 2-2}$$

The data are divided into two sub-arrays of equal size (k=0, ... , N/2-1) containing the even and odd parts of the original array. The process is repeated until the sub-arrays contain only one point, the Fourier transform of a single point is itself.

The process is iterative, each transform is calculated from the transforms of the component sub-arrays. Equation 2-2 can be re-written as:

$$\overline{X}_k = A_m + W^k B_m \qquad \text{and}$$

$$\overline{X}_{k+N/2} = A_m - W^k B_m \qquad \text{Equation 2-3}$$

This equation is known as the FFT butterfly because of the symmetry of the resulting data flow graph.

The resulting "in-place" algorithm for the computational kernel is:

$$Y_k = Y_k + W^k Y_{k+N/2} \qquad \text{and}$$

$$Y_{k+N/2} = Y_k - W^k Y_{k+N/2} \qquad \text{Equation 2-4}$$

where Y is the N element data array to be transformed. When calculating in-place, the results are stored back over top of the original numbers and no explicit odd-even sorting is performed. Doing this reduces the amount of storage required to calculate the transform. This algorithm requires the initial data to be pre-sorted into digit reversed order before applying the FFT so that the resulting data will be in correct sequence.

Digit reversal refers to the process whereby each element of a data array is exchanged with the element that has the same digits in its index but in reverse order. The process depends upon the overall size of the array and the base representation of the index. For base 2, i.e. radix–2 FFT, the process is accomplished by reversing the order of the bits in the binary representation of the index, hence the name "bit-reversal".

The time required to compute an FFT is roughly proportional to $N \log_2 N$, discounting bit-reversal time. The relative time required to calculate a transform of size N using the two Fourier transform algorithms is given by the following equation:

$$\frac{DFT}{FFT} = \frac{N^2}{N \log_2 N} = \frac{N}{\log_2 N} \qquad \text{Equation 2-5}$$

This approximation is valid because execution time is dominated by floating point operations. Even for not very large values of N the time savings from the FFT algorithm can be considerable and this is what makes Fourier transform spectroscopy really practical for routine analysis.

The Cooley–Tukey FFT algorithm is the most commonly used Fourier transform algorithm in use today but others exist, the most notable ones are the prime factor algorithm and the fast Hartley transform. The performance of both of these new algorithms along with variations of the Cooley–Tukey algorithm are compared with the original Cooley–Tukey radix–2 implementation.

2-1. The Cooley–Tukey FFT algorithm

Execution time for the general Cooley–Tukey algorithm is actually proportional to N $\log_m$ N where m is the radix (base) of the algorithm. Therefore the obvious way of improving performance is to increase the value of m. Base 4 and 8 algorithms have been described in the literature [39] but the algorithms for higher values rapidly become more complex.

For higher bases the data is divided into a larger number of sub-arrays, this requires fewer passes through the data. The result is that the inner-most loop (known as the butterfly) computes more points in a pass. The algorithms for higher bases are particularly attractive for parallel processing since each point calculated by the butterfly can be computed simultaneously on separate processors. The following is the butterfly for a radix-4 decimation in time algorithm.

$$\begin{aligned}
Y_k &= Y_k + W^k Y_{k+N/4} + W^{2k} Y_{k+N/2} + W^{3k} Y_{k+3N/4} \\
Y_{k+N/4} &= Y_k + j W^k Y_{k+N/4} - W^{2k} Y_{k+N/2} - j W^{3k} Y_{k+3N/4} \\
Y_{k+N/2} &= Y_k - W^k Y_{k+N/4} + W^{2k} Y_{k+N/2} - W^{3k} Y_{k+3N/4} \\
Y_{k+3N/4} &= Y_k - j W^k Y_{k+N/4} - W^{2k} Y_{k+N/2} + j W^{3k} Y_{k+3N/4} \qquad \text{Equation 2-6}
\end{aligned}$$

The disadvantage of using bases higher than 2 is that they reduce the number of sizes of transform available to the user. In addition, the real performance benefits of higher bases are also less than expected on single processor computers.

2-2. The Prime factor algorithm

In the mid 70's, various prime factor algorithms (PFA) were derived for calculating Fourier transforms [40]. PFA's require the size of the transform (N) be the product of a series of prime numbers (Equation 2-7), hence the name, and performs the actual calculations as a series of short cyclic convolutions.

$$N = N_1 \times N_2 \ldots \times N_d \qquad \text{Equation 2-7}$$

This is made possible by re-mapping the N points into a d dimensional matrix of size $N_1, N_2, \ldots, N_d$.

The most notable of the prime factor algorithms is the Winograd Fourier transform algorithm (WFTA) [41, 42, 43] which achieves high theoretical performance by reducing the number of multiplications required (at the expense of some additions) to calculate the transform.

In actual fact, a Fortran IV implementation of the WFTA on a DEC PDP–11/10 was only about 30% faster than a comparable FFT. The degradation in performance over that predicted by theory is due to the large amount of overhead in the algorithm. Much of this is due to the extensive pre- and post-sorting of the data required by the WFTA [44].

In addition, the WFTA for each size of data array must be custom coded because there is no single general algorithm for the short cyclic convolutions used by the WFTA. The permissible sizes for data arrays are also restricted to numbers composed of mutually prime factors.

I have concluded that the WFTA is unsuitable for FTS applications because the speed benefit does not offset its complexity and the loss of flexibility due to the lack of a complete general algorithm. This conclusion is also supported by Morris [44] who has also reported cases where the WFTA is slower than the FFT.

2-3. The Hartley transform

An alternate means to calculating the Fourier transform is via a Hartley transform [45]. The discrete Hartley transform (DHT) [46], shown in Equation 2-8, is similar in form to the DFT and also requires N^2 operations for a transform of length N.

$$H_k = \sum_{m=0}^{N-1} h_m \operatorname{cas}(2\pi mk/N) \quad \text{where} \quad k = 0, \dots, N-1$$

$$\operatorname{cas}(x) = \cos(x) + \sin(x) \qquad \text{Equation 2-8}$$

The major advantage of the DHT over the DFT is that it is a real transform instead of complex, this results in considerable savings in execution time when dealing solely with real data.

Bracewell [47] has derived a fast ($t<N^2$) algorithm for the Hartley transform shown in Equation 2-9. The fast Hartley transform (FHT) algorithm, like the FFT, results from successively factoring the data and also requires an index reversal pre-sort.

$$Y_k = Y_k + Y_{k+N/2}\cos(2\pi k/(N/2)) + Y_{N-k}\sin(2\pi k/(N/2)) \text{ and}$$

$$Y_{k+N/2} = Y_k - Y_{k+N/2}\cos(2\pi k/(N/2)) - Y_{N-k}\sin(2\pi k/(N/2)) \qquad \text{Equation 2-9}$$

It should be noted that the Y_{N-k} term introduces an asymmetric factor into the algorithm.

The Hartley transformed data is easily converted to the corresponding Fourier transform with the following relationship [46]:

$$F_k = \frac{(H_k + H_{N-k})}{2} - j\frac{(H_k - H_{N-k})}{2} \qquad \text{Equation 2-10}$$

When only a power spectrum is desired the preceding equation can be simplified to:

$$P_k = \frac{H_k^2 + H_{N-k}^2}{2} \qquad \text{Equation 2-11}$$

Despite the inherent mathematical advantages of the Hartley transform, the FHT has slightly more processor overhead than the FFT. This is partly due to the asymmetric nature of the FHT which prevents the transform from being calculated in place. A temporary array containing the results of the previous pass is required, eliminating the memory savings of a real transform. In addition, the results of the Hartley transform must be converted to the corresponding Fourier transform.

Overall, the FHT is about 25% faster than the FFT with no advantage with regards to memory usage. My implementation of the FHT on a Macintosh Plus is considerably more efficient than that recently published in Byte [48]. Where-as the published program computed a 512 point transform in 27 s my routine will do a 1024 point amplitude spectrum in 23 s and my

FFT routine will do 1024 points in 29 s. These times were all recorded on a Macintosh Plus using 80 bit math performed in software. I use the same TML Pascal compiler as used in the article and my routines are highly optimized.

2-4. Optimizing the Cooley–Tukey algorithm

After implementing and benchmarking the various methods for calculating Fourier transforms it was decided to continue using the original Cooley–Tukey radix–2 algorithm, no newer algorithm delivered any dramatic increases in performance. Despite the minor performance penalties, this algorithm is the most general purpose and is the easiest to implement. The question now is how to get the most out of the FFT algorithm.

The performance of an FFT routine can be improved significantly if the algorithm is optimized for execution on a specific computer. Two types of optimizations are possible, general ones which modify the algorithm to make more efficient use of computer resources and implementation specific ones which take advantage of the features of the specific processor.

The FFT algorithm can be thought of as three nested loops (see Figure 2). The outer-most loop determines the current sub-array size and iterates through all possible sizes from 2 to N. The middle loop divides the entire data array into sub-arrays and loops through them. Note that these are not the sub-arrays described in Equation 2-2 because they are not sorted even/odd and are twice the required size. Finally the inner-most loop operates on each sub-array to calculate its FFT using the butterfly equations (Equation 2-3).

The two outer loops factor the original data to the point where the Fourier transform is trivial, i.e. one point, and use the result of the previous

```
{ $N = 2^M$ }
{ $W(x) = e^{-j2\pi x/N}$ , $j = \sqrt{-1}$ }

FOR I := 1 TO M
   BEGIN;
   L := 2**I;
   L2 := L/2;
   NDL := N/L;
   FOR J := 0 TO (N-1) BY L
      FOR K :=0 TO (L2-1)
         BEGIN
         T := X[J+K+L2]*W(NDL*K);
         X[J+K+L2] := X[J+K]-T;
         X[J+K] := X[J+K]+T;
         END;
   END;
```

Figure 2. Pseudo-code for a direct implementation of the Cooley–Tukey radix–2 decimation in time algorithm. All non-integer operations are complex.

transforms to calculate the next larger transform. If the starting data was index digit reversed prior to executing the three loops, then the result is the Fourier transform of the original data. This avoids having to perform an even/odd sort every time a transform is calculated.

As the middle loop cycles through each sub-array in the overall data set the inner loop calculates the same set of exponential weighting factors over and over again, this results in a lot of wasted time since we are continually recalculating the same numbers. The obvious way to eliminate this redundancy is to pre-calculate all the required exponential factors and save them in an array, where they can be accessed from the inner loop, before entering the middle loop.

Unfortunately, this solution requires that enough space be reserved for all the factors required by the largest sub-array. An alternate solution to this problem is to re-arrange the algorithm so that redundant calculations are eliminated. This can be achieved by switching the two inner loops (Figure 3) so that all the butterflies that require a given exponential factor are computed first before calculating a new exponential factor. This results in substantial time savings without having to resort to look-up tables.

Further reductions in execution time result from fine tuning the implementation of the algorithm for execution on digital computers. This is done by replacing time intensive operations with more economical ones wherever possible. For instance, multiplication, division, and exponentiation are expensive operations while addition and bit shifts are not.

The next step is therefore the removal of as many multiplication, division, and exponentiation operators as possible, especially from the inner

```
{ $N = 2^M$ }
{ $W(x) = e^{-j2\pi x/N}$ , $j = \sqrt{-1}$ }

FOR I := 1 TO M
   BEGIN;
   L := 2**I;
   L2 := L/2;
   NDL := N/L;
   FOR K := 0 TO (L2-1)
      BEGIN
      V := W(NDL*K)
      FOR J :=0 TO (N-1) BY L
         BEGIN
         T := X[J+K+L2]*V;
         X[J+K+L2] := X[J+K]-T;
         X[J+K] := X[J+K]+T;
         END;
      END;
   END;
```

Figure 3. A more efficient implementation of the Cooley–Tukey radix-2 decimation in time algorithm obtained by reversing the two inner loops.

loops. The obvious places for improvements are the variables that keep track of the sub-array parameters, they are recalculated with each iteration of the I loop, it makes the algorithm clearer but slows down execution. By keeping track of the current sub-array size there should be no need to do this, the arrays always double in size with each iteration.

Instead of recalculating L, the size of the current sub-array, with each pass of the outer loop it is more expedient to simply double its value. The most efficient way of doubling a number on a computer is to add it to itself. By assigning L2 (half of the current value of L) to the previous contents of L, an entire calculation is eliminated. Since the sub-arrays always double in size the number of sub-arrays, NDL, will be half the previous value.

In implementing these changes, one exponentiation and two divisions are replaced by one addition and a division. The only penalty is the algorithm is no longer as readable as it was before. The same principles can also be applied to the increment for the exponential factor (NDL*K), an addition is substituted for a multiplication.

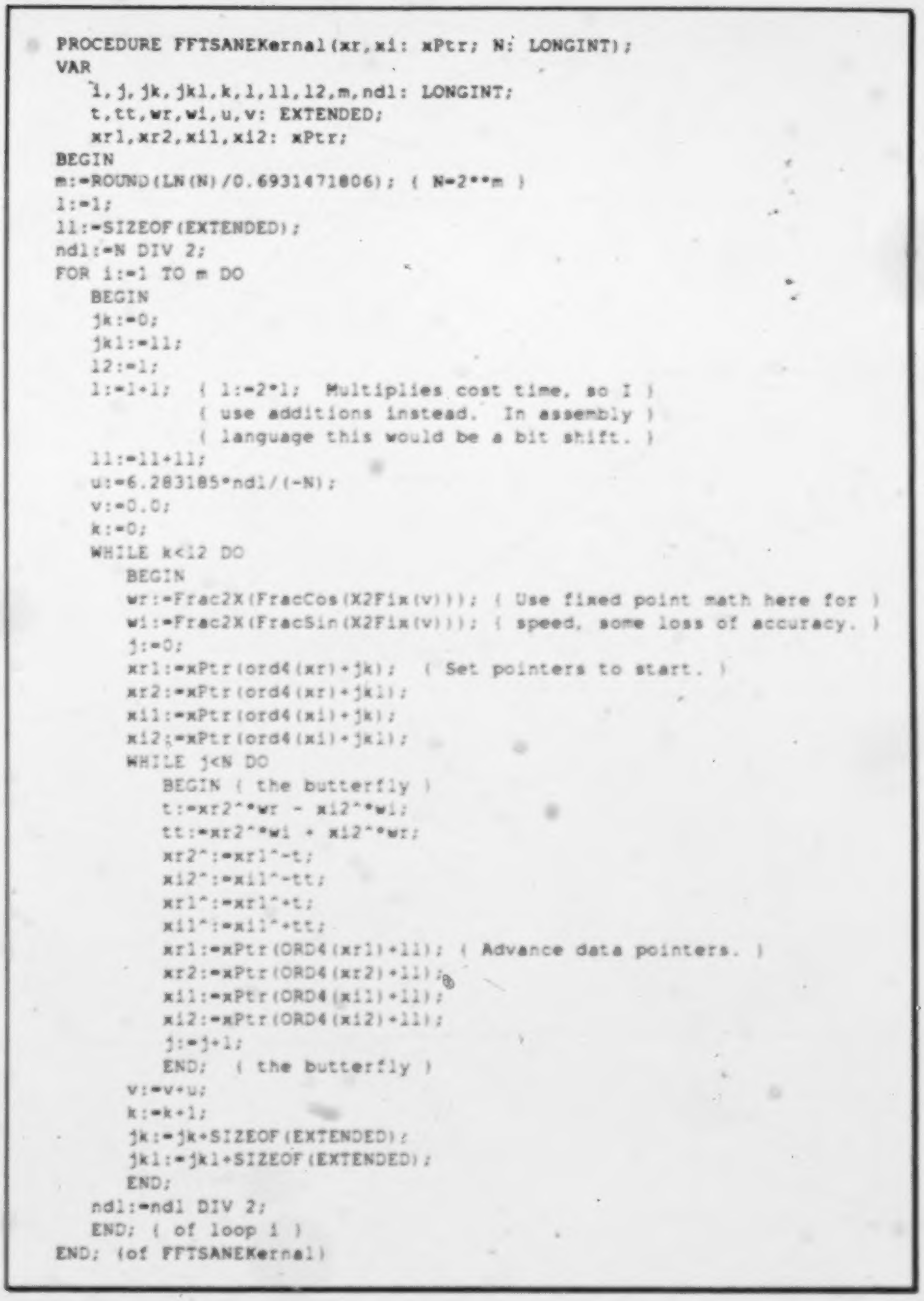
The final implementation of the FFT algorithm is shown in Figure 4 and the actual Pascal source code in Figure 5. Note that the Pascal code breaks the complex operations down to their real components using inline code. The performance overhead associated with implementing complex operators using generalized subroutines are prohibitive when called within large nested loops.

The Pascal code also uses direct pointer manipulation to access the data instead of array indexing as shown in the pseudo-code. This was done for two reasons, using pointers is faster than array indexing and the compiler used

```
{ $N = 2^M$ }
{ $W(x) = e^{-j2\pi x/N}$ , $j = \sqrt{-1}$ }

L := 1;
NDL := N/2;
FOR I := 1 TO M
    BEGIN;
    L2 := L;
    L := L+L;
    U := 0;
    FOR K := 0 TO (L2-1)
        BEGIN
        V := W(U);
        FOR J :=0 TO (N-1) BY L
            BEGIN
            T := X[J+K+L2]*V;
            X[J+K+L2] := X[J+K]-T;
            X[J+K] := X[J+K]+T;
            END;
        U := U+NDL;
        END;
    NDL := NDL/2;
    END;
```

Figure 4. FFT algorithm optimized for execution on digital computers.



```
PROCEDURE FFTSANEKernal(xr,xi: xPtr; N: LONGINT);
VAR
   i,j,jk,jk1,k,l,l1,l2,m,ndl: LONGINT;
   t,tt,wr,wi,u,v: EXTENDED;
   xr1,xr2,xi1,xi2: xPtr;
BEGIN
m:=ROUND(LN(N)/0.6931471806); { N=2**m }
l:=1;
l1:=SIZEOF(EXTENDED);
ndl:=N DIV 2;
FOR i:=1 TO m DO
   BEGIN
   jk:=0;
   jk1:=l1;
   l2:=l;
   l:=l+l;  { l:=2*l;  Multiplies cost time, so I }
            { use additions instead.  In assembly }
            { language this would be a bit shift. }
   l1:=l1+l1;
   u:=6.283185*ndl/(-N);
   v:=0.0;
   k:=0;
   WHILE k<l2 DO
      BEGIN
      wr:=Frac2X(FracCos(X2Fix(v))); { Use fixed point math here for }
      wi:=Frac2X(FracSin(X2Fix(v))); { speed, some loss of accuracy. }
      j:=0;
      xr1:=xPtr(ord4(xr)+jk);  { Set pointers to start. }
      xr2:=xPtr(ord4(xr)+jk1);
      xi1:=xPtr(ord4(xi)+jk);
      xi2;=xPtr(ord4(xi)+jk1);
      WHILE j<N DO
         BEGIN { the butterfly }
         t:=xr2^*wr - xi2^*wi;
         tt:=xr2^*wi + xi2^*wr;
         xr2^:=xr1^-t;
         xi2^:=xi1^-tt;
         xr1^:=xr1^+t;
         xi1^:=xi1^+tt;
         xr1:=xPtr(ORD4(xr1)+l1); { Advance data pointers. }
         xr2:=xPtr(ORD4(xr2)+l1);
         xi1:=xPtr(ORD4(xi1)+l1);
         xi2:=xPtr(ORD4(xi2)+l1);
         j:=j+l;
         END;  { the butterfly }
      v:=v+u;
      k:=k+1;
      jk:=jk+SIZEOF(EXTENDED);
      jk1:=jk1+SIZEOF(EXTENDED);
      END;
   ndl:=ndl DIV 2;
   END; { of loop i }
END; {of FFTSANEKernal}
```

Figure 5. Pascal source code for Macintosh FFT.

doesn't allow arrays of greater than 65,536 bytes in size. The symmetry of the FFT butterfly also facilitates the use of pointers. The user type "xPtr" is a pointer to an 80 bit extended precision floating point value (see Figure 6).

The actual code uses the identity $e^{jx} = \cos(x) + j \sin(x)$ to calculate the exponential factor. The code shown is for a Macintosh Plus, which performs floating point operations in software, and 32 bit fixed point math is used to calculate the trig values. This increases speed immensely but at the cost of accuracy. The Macintosh II version of the FFT routine uses 80 bit extended precision floating point values.

The ultimate form of optimization is to implement the FFT algorithm in assembly language. This way compromises made by language compilers can be eliminated and all the power of the central processor can be used. Assembly language implementations of the FFT are twice as fast as the equivalent high level language versions. This has been confirmed on both IBM-XT and Macintosh II computers (see Table 1).

Note that the Macintosh II does not appear to be much faster than the IBM-XT; this is misleading because the two are running different implementations of the FFT algorithm. Because of the memory architecture of the Intel 8088 CPU the IBM version of the FFT had no reason to conserve memory and therefore used pre-calculated look-up tables of trig constants. The time required to generate these tables are not included and will add 9 seconds to the IBM times in Table 1 (these tables were always generated in a high level language).

```
TYPE
   xPtr = ^EXTENDED;
```

Figure 6.   Pascal definition for the xPtr type.

	IBM–XT	Macintosh II
high level language	22 s	50 s
assembly language	9.5 s	20 s

Table 1. Comparison of high level language and assembly language implementations of FFT algorithm. IBM times are for a 4096 point transform while the Macintosh times are for a 16384 point transform. Math coprocessors and non-optimizing compilers were used on both machines. MS Fortran 3.13 and TML Pascal 2.5 used on IBM and Macintosh respectively.

The Macintosh II doesn't use look-up tables at all but calculates all the required trigonometric values on demand. This introduces a lot of computational overhead but saves a lot of memory. Saving memory on the Macintosh II is very important since multiple data sets are stored in memory and concurrent tasks may also require memory.

Another factor that affects relative performance between the two machines is operating system overhead. MS-DOS is a relatively simple, single task operating system which steals very little time from the current user task. The Macintosh operating system, on the other hand, is a sophisticated event driven graphics operating system which supports background tasks that steal time from the current task. For a truly fair comparison between the two types of computers, the IBM should be running the Windows operating system.

It should also be mentioned that all floating point operations were performed using 80 bit precision floating point math. This was because both machines were equipped with hardware floating point coprocessor chips. Both the Intel and the Motorola math chips work internally with 80 bits, therefore there is no speed advantage in using a lower precision.

The use of assembly language allows the implementation to be tailored for a specific processor. Intermediate results and address pointers can be kept in registers, eliminating time consuming memory accesses. In addition, memory is accessed more efficiently since all the processor addressing modes can be used directly.

On the Macintosh II, assembly language also allows direct access to the floating point coprocessor and its specialized instruction set. An example is the coprocessor instruction FSINCOS which returns both the sine and cosine of an input argument in two registers. This results in a 40% time savings over calculating them separately [49]. The MPW assembler source listing for the Macintosh II assembly language FFT is in Appendix D.

Because the FFT is dominated by the time it takes to perform the necessary floating point operations, there is a practical limit to optimizing it. At one stage, several instructions were re-coded to make the entire routine 26 bytes smaller and it was calculated that 8 million clock cycles would be saved when performing a 16K transform. Unfortunately, with a 16 MHz clock, this translates to a half second savings which is exactly what was observed when the routine was executed. This is the reason why an assembly language FFT was not implemented for machines without math coprocessors.

The Pascal source code for the bit reversal routine is shown in Figure 7, it was adapted from Eckhouse and Morris [50]. The same Pascal routine is used with both the Pascal and assembly language FFT routines. An iterative method in which the new index is calculated from the previous is used to save time. Also note that some symmetry exists, the indices for the second half of the array are easily computed from those in the first half, this halves the computational time.

The Macintosh FFT routine has been modularized so that its interface is machine independent. If floating point hardware is available then an assembly language routine is used, otherwise a Pascal routine is used. The overall FFT routine is shown in Figure 8, it determines the hardware environment at runtime and calls the appropriate routines to actually calculate the FFT.

```
PROCEDURE BitReverse(xr,xi: xPtr; N: LONGINT);
VAR
   i,j,k,l,m,n2,off1,off2,off3: LONGINT;
   t: EXTENDED;
   xr1,xr2,xi1,xi2: xPtr;
BEGIN
n2:=N DIV 2;
off3:=n2*SIZEOF(EXTENDED);
j:=0;
FOR i:=0 To n2-1 DO
   BEGIN
   k:=i;
   off1:=k*SIZEOF(EXTENDED);
   off2:=j*SIZEOF(EXTENDED);
   IF k<j THEN
      BEGIN
      xr1:=xPtr(ORD4(xr)+off1);
      xr2:=xPtr(ORD4(xr)+off2);
      t:=xr1^;
      xr1^:=xr2^;
      xr2^:=t;
      xi1:=xPtr(ORD4(xi)+off1);
      xi2:=xPtr(ORD4(xi)+off2);
      t:=xi1^;
      xi1^:=xi2^;
      xi2^:=t;
      END;
   { Do second half of arrays }
   k:=k+n2;
   l:=j+1;
   off1:=off1+off3;
   off2:=off2+SIZEOF(EXTENDED);
   IF k<l THEN
      BEGIN
      xr1:=xPtr(ORD4(xr)+off1);
      xr2:=xPtr(ORD4(xr)+off2);
      t:=xr1^;
      xr1^:=xr2^;
      xr2^:=t;
      xi1:=xPtr(ORD4(xi)+off1);
      xi2:=xPtr(ORD4(xi)+off2);
      t:=xi1^;
      xi1^:=xi2^;
      xi2^:=t;
      END;
   { calculate next index }
   m:=n2;
   WHILE j>=m DO
      BEGIN
      j:=j-m;
      m:=m div 2;
      END;
   j:=j+m
   END;
END; {of BitReverse}
```

Figure 7. Pascal source code for bit reversal routine.

```
{ The assembly language FFT routine has to be linked in. }
PROCEDURE FFTasmKernal(xr,xi: xPtr; N: LONGINT); EXTERNAL;

PROCEDURE FFT {(xr,xi: xPtr; N: LONGINT; inverse: BOOLEAN)} ;
VAR
   i: LONGINT;
   xr1,xi1: xPtr;
   err: OSErr;
   myWorld: SysEnvRec;
   h: Handle;

BEGIN
err:=SysEnvirons(1,myWorld);  { Get hardware parameters. }
IF inverse THEN   { Change sign of imaginary if IFFT. }
   BEGIN
   xi1:=xi;
   for i:=0 to N-1 do
      BEGIN
      xi1^:= -xi1^;
      xi1:=xPtr(ORD4(xi1)+SIZEOF(EXTENDED));
      END;
   END;

{ sort the arrays first }
BitReverse(xr,xi,N);

{ do FFT }
IF myWorld.hasFPU THEN
   FFTasmKernal(xr,xi,N)
ELSE
   FFTSANEKernal(xr,xi,N);

IF inverse THEN          { Scale results if IFFT. }
   BEGIN
   xr1:=xr;
   xi1:=xi;
   FOR i:=0 TO N-1 DO
      BEGIN
      xr1^:=xr1^/N;
      xi1^:=xi1^/N;
      xr1:=xPtr(ORD4(xr1)+SIZEOF(EXTENDED));
      xi1:=xPtr(ORD4(xi1)+SIZEOF(EXTENDED));
      END;
   END;
END; {of FFT}
```

Figure 8. The complete Macintosh FFT routine.

2-5. Conclusions

It is apparent that advances in Fourier transform algorithms will not yield any improvements that will have a significant impact on the application of Fourier techniques. Orders of magnitude improvements in performance are still required in order to make Fourier methods "real-time". This is not to say that the search for more efficient algorithms should be halted, but rather one should not expect miracles from the mathematicians.

The only way to achieve significantly faster transforms is through improvements in the computing hardware on which the programs run. Faster computers are not the solution, very soon now the speed of electronic circuits will approach their quantum and relativistic limits. Parallelism is the answer, it allows many orders of magnitude improvements using existing technologies [50].

Fortunately, the FFT algorithm is highly amenable to parallel implementation due to its highly symmetric nature. Both tightly coupled and massively parallel systems can be used effectively to compute FFT's. With only a few processors the algorithm can be expressed so that each computes a part of the butterfly. Substantial savings result because the entire butterfly can be computed simultaneously. When many processors are available they can be organized into small groups that are used to calculate many individual butterflies concurrently. With enough processors, several orders of magnitude improvement in execution time could result.

Chapter 3

Implementing new concepts in user interfaces.

Computers have become an essential part of chemical research for both data acquisition and performing calculations. While the power of computing hardware available to the chemist has increased steadily over the years the same cannot be said for the software that runs on said computers, especially in the area of human processor interfaces [52].

In the early years, researchers had to write their own software and were happy if their programs even ran correctly. Because of their highly specialized requirements, researchers today continue to write their own software in order to supplement the few commercially available packages. Despite the advances in software engineering, programs written by researchers continue to be less than the state of the art. They are usually large, buggy, and not very elegant, but they do work after a fashion.

The reason for this continuing trend is the fact that programming is a highly specialized field in itself and few researchers can devote the time and effort to both their research and mastering the art of programming. Also, researchers rarely have unlimited access to the services of a professional programmer.

The lack of commercial programs is due to the reverse, programmers are not aware of the special needs of researchers. This is in addition to software's low priority when allocating funds which makes developing products targeted towards researchers a risky proposition for software developers. Because of these factors, the software that is available is very expensive due to the narrowness of the vertical market.

Spending the time and effort to develop a proper user interface is no longer a luxury [53]. Poorly implemented programs inhibit productivity and creativity. The user wastes time working around the software and it can break up the flow of thought when interpreting results. Also counter-productive, is the writing of new programs or the constant modification of existing ones to perform new functions.

Analytical chemists and instrumentation companies are just starting to realize the importance of the user interface. Karanassios and Horlick [54] have presented a proposed user interface tailored for ICP–OES and Barnett [55] discusses some of the basic concepts and principles involved in user interface design. Barnett also comments on some specific requirements that are important to analytical chemists.

In order to illustrate the concepts presented in this chapter, I have written a program called SpectroPlot that interfaces the Fourier transform spectrometer in our laboratory to a Macintosh II personal workstation. This program was also used to acquire all the data for this thesis.

SpectroPlot was written using TML Pascal from TML Systems, Mac C assembler from Consulair Corporation, and ResEdit from Apple Computer. The source code consists of >10,000 lines of poorly documented code that occupies more than a quarter million bytes of disk storage. Because of space considerations, the source code is not included in this thesis. I mention the fact that the source code is poorly documented because once I document it adequately it should grow to twice its present size.

3-1. Design paradigms

In designing a user interface that facilitates rather than hinders scientific research, the latest advances in software engineering should be used. This is in addition to improved design principles and never forgetting that research has special needs.

Foremost in user interface design is the role of interactivity. Many interfaces claim to be interactive but are in fact question and answer sessions in which the user must respond to the computer rather than the other way around. Human beings are multi-modal creatures capable of simultaneous thoughts and actions. In order for a user interface to be truly interactive, it must also be multi-modal [56].

A new concept in user interfaces is visualization, that is presenting information visually. Scientists have used this concept for many years to make large data sets more manageable and complex ideas more comprehensible [53]. For example, it is much easier to interpret a plot of a spectrum than a table of intensities at each wavelength. Visualization has gained more importance as scientific instruments gather more data at faster rates, for without it the researcher is soon overwhelmed by the sheer mass of data [57].

When visualization is coupled with a set of self consistent rules to produce an analog of an environment what is created is a "virtual reality" [52, 53, 58]. The rules governing the environment need not be real physical ones, purely artificial rules are acceptable. It is only important that the rules be self consistent. If they are not then the environment will be perceived as unnatural. Virtual reality can be applied to produce a user interface that is both graphical in nature and transparent to the user.

Graphic user interfaces are easier to use than text based ones because the human brain is unmatched when it comes to recognizing visual patterns. Visual bandwidth is much higher than textual bandwidth, hence the old saying "a picture is worth a thousand words." Text requires further processing in order to convert images of letters into actual words. Text interfaces also force the user into the unnatural mode of memorizing text commands [52, 59]. Reading and typing long command strings is extremely slow while shorter strings can be very cryptic and require a longer training period.

Virtual realities and graphic interfaces are no guarantee of ease of use. In order for a user interface to be easy to use, its virtual reality model should be based on some real world metaphor that is familiar to the user [60]. The most successful graphic user interface to date is the "desktop" metaphor pioneered by Xerox PARC (Palo Alto Research Center) on their Alto and Star machines [61], and first introduced to the general public by Apple Computer on their Macintosh series of computers. The desktop metaphor is one implementation of a "spatial data management system" proposed by Negroponte at the Architecture Machine Group at MIT [59].

In this type of interface a pointing device, usually a mouse, which controls an on-screen pointer is used in conjunction with on-screen graphical objects (icons, controls, and pull-down menus) to input commands [62, 63]. In order to use a computer running under a desktop metaphor the user need only master a few simple mechanical operations such as: clicking, double clicking, and dragging. No major mental exercises are required and the physical feedback from moving the pointing device reinforces the actions shown on the screen [59].

The desktop metaphor is also a multi-modal environment and many implementations are even multi-tasking. The interface is event driven, responding to user actions rather than demanding them. The user can select any of the objects present on the screen, the different object types will respond immediately and in a consistent manner. Even hierarchical menus can be traversed instantly instead of going through the multiple levels found in earlier menu driven interfaces.

The desktop metaphor has been so successful that all the major computer manufacturers are implementing it on their machines. It has also been and is currently the object of several lawsuits [64]. The only drawback to this type of user interface is the amount of processing and code overhead required to support it. A sixteen bit microprocessor running at 8 MHz with 512 kilobytes of memory is the minimum hardware requirement but with the latest generation of 32 bit machines this is no longer a major factor.

Another important design factor is synergy, or "Why re-invent the wheel?" Many commercially available programs, while not geared specifically towards the needs of researchers, can be used or adapted for research [65]. Such programs include computer aided design, database managers, outliners, presentation graphics, project management, spreadsheets, and word processors. Spreadsheet programs are the most commonly used type of commercial applications because of their analysis, report, and simulation capabilities [66].

Salit [67] has recently discussed the advantages, limitations, and salient features of spreadsheets with respect to analytical chemistry.

Instead of combining elements of these disparate program types into a single massive program, it is better to write a program that builds upon the abilities of the various commercial programs. This makes for a smaller, more efficient program and better use of the programmer's time because (s)he can draw on the skills of other programmers. This can be accomplished by writing a program that can exchange data with other programs, preferably using some standard format or through co-resident programs such as Macintosh desk accessories.

Take this thesis for example; it was produced using a number of programs on an Apple Macintosh. Equation formatting was done using the Expressionist desk accessory, charts with Trapeze, and the figures were assembled with MacDraw II. All the resulting elements were combined with the body text using Microsoft Word and then directly printed out on a laser printer. Absolutely no manual paste-up was performed. This was made possible by the standard data formats for graphics and text and data exchange protocols found on Macintosh computers.

This allows the user to choose the best program for each job. Integrated packages that combine several functions into a single program are usually laden with compromises. Because of these compromises, there is always a stand-alone program that is superior to each of the modules in an integrated package [68]. Integrated packages are normally found in environments that don't facilitate inter-program data exchange, such as MS-DOS.

This is the reason why scientific word processors do not exist for the Macintosh [69]. All Macintosh word processors are capable of importing and displaying graphics from specialized programs, therefore all the user needs, in addition to a normal word processor, is whatever specialized software is required for their purposes (i.e. drawing chemical structures).

Other advantages of using commercial non-scientific software is its lower cost, since it is sold to a larger market, and the ability to spread the purchases out over time. These factors can reduce the impact on one's budget.

It is also beneficial if all the programs employed by a user function in a similar manner and have the same user interface. This leads to the concept of consistency [70]. Consistency allows the user to quickly learn how to use new programs. Once the user has learned to use one program, (s)he can use all others that share the same user interface. This is a particularly important factor when one considers the number of software packages that exist.

Consistency also implies standardization which confers stability to a program. Programs that adhere to standards are less likely to crash in the future. Standards also allow a program to be more configuration independent by guaranteeing a specific level of system services. Good standards should allow for future expansion in an orderly manner, this allows programs to benefit from future developments. Without the ability for future expansion the standard will be incapable of growth and will eventually wither and die.

A good example is the standard floating point library built into the Macintosh. The version of this library that exists on Macintosh II's use the floating point coprocessor instead of software routines to perform floating point operations. This means that the previous generation of programs that used the standard library are able to run on the new Macintoshes and enjoy the benefits of the new hardware without having to be re-written.

Finally, the user interface should be clear and concise, there should be no extra frills, and every part should be functional. Features should not be added haphazardly but should fit smoothly into the overall scheme of things. This reduces visual clutter which can prove confusing to novice users and conveys meaning more rapidly. However, beware of over simplifying; meaning can be lost entirely or require more time for interpretation.

Time spent properly designing a program and its user interface is never wasted. This is a problem with most programs written by scientists. They are minimally designed to perform a certain calculation with little thought to overall form. They are continually modified, not always by the same person, and extra functions may be added on as an afterthought. The resulting program becomes overly large and convoluted. The program functions, but is hard to learn and use, and is sometimes impossible to learn without going over the source code.

3-2. Interacting with data

A good user interface is transparent to the user. The ultimate goal is for users to be able to interact with and manipulate data without realizing that they are using a computer. The use of a pointing device greatly facilitates easy interaction with data by making it possible to directly point at data and objects on the screen. The combination of mouse, high resolution graphics, virtual windows, and pull down menus form the core of our user interface.

Each data set appears in its own window and all data windows are moveable and resizable. The windows can be stacked to conserve screen area or if multiple monitors are available can be distributed arbitrarily among them. Windows can be copied and sub-windows can be created. Sub-windows allow the user to simultaneously view in detail several different portions of a single data set. Sub-windows constitute a separate independent view of a single data set, and all sub-windows will be adjusted accordingly if the data is changed.

A subset of the full data can also be extracted. This allows the user to concentrate on the region of interest without having to load the entire data set into memory. This is important because many operations work on the entire data set, thereby wasting a lot of time if much of the data is not of interest.

There is no limit to the number of windows that can be simultaneously open. All currently open data sets are kept in memory to minimize access time but this limits the amount of data that can be open to the amount of available memory.

Each window can also be independently calibrated in arbitrary units which allows the user to view the data using units that have real physical meaning. This is an easy feature to implement and is not a new concept as it is used in many commercial programs.

SpectroPlot allows the user to directly select any data displayed on the screen using a mouse. It is not necessary for the user to enter "data selection mode", whenever the mouse cursor is moved over the graph region of the active data window it changes from an arrow to a cross-hair to indicate that data can be selected (see Figure 9a). Any window can be made active by simply clicking on any portion of it.

Selections are made either by clicking on a single data point or by dragging over a region. The selection is highlighted so that it can be seen (see Figure 9b). This is made possible by keeping track of the portion of each data set that appears in the window and mapping it to the window coordinate space. The data point under the cursor can then be calculated from the window position by reversing the mapping process.

This ability to directly access the data without requiring any intervening modes forms the basis for the user interface that I have implemented. It is a direct implementation of the "see and point" paradigm [69] which combines interactivity with visualization.

Another ability of SpectroPlot is that of being able to scroll through the data when only a subset is visible in the window. This ability allows rapid detailed review of the data and facilitates searching. Scrolling changes the visible portion of the data set and re-maps the data to the window. To keep scrolling smooth and fast only the newly visible data is actually drawn, the screen representation of the rest of the data is simply shifted over.

The selected data forms the basis for several operations, for example calculations on the selected data or creation of new windows containing only the selected data. Selections are also a rapid means of viewing a specific sub-

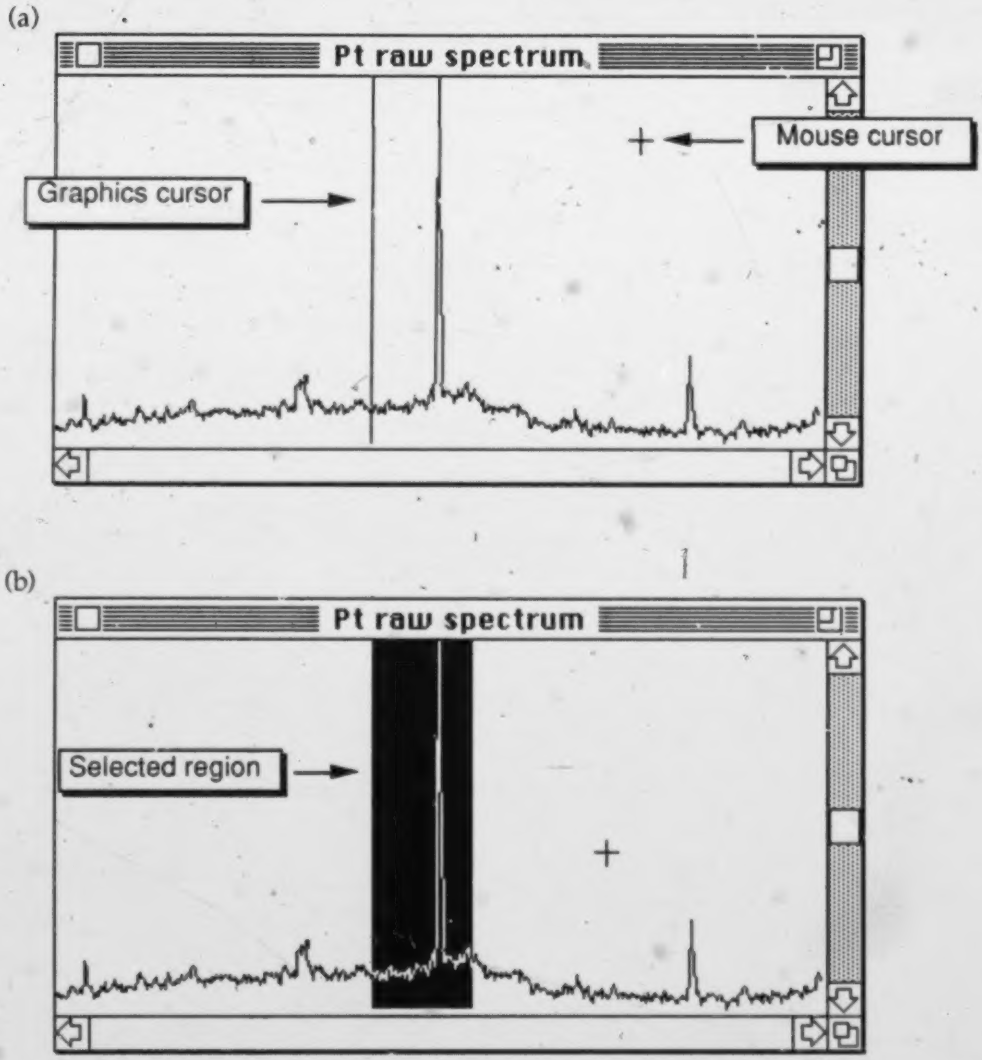

(a)
Pt raw spectrum
Graphics cursor
Mouse cursor
(b)
Pt raw spectrum
Selected region

Figure 9. Data windows from SpectroPlot showing the various cursors (a) and a spectrum with a selected region (b).

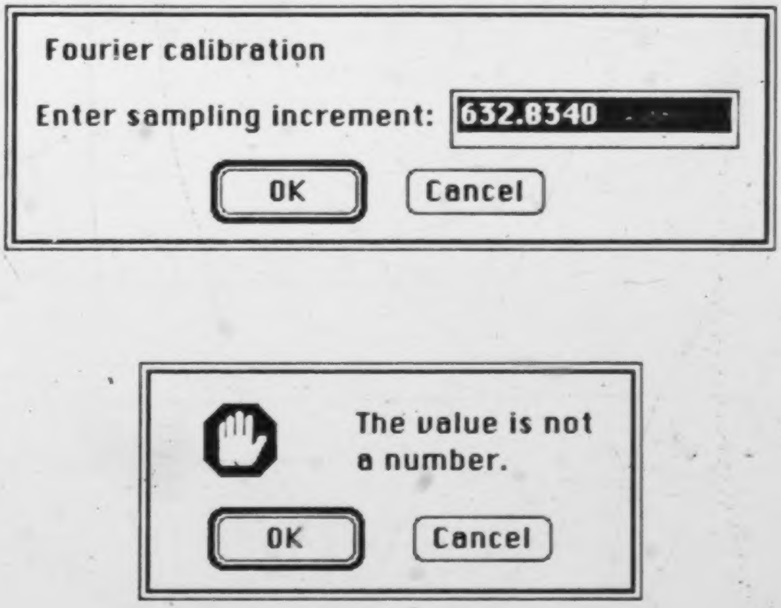
set of the overall data set; double clicking a selection will automatically redraw the window so that the selection fills the entire window.

The use of modes is avoided as much as possible but cannot be completely eliminated. This is accomplished by making the most common on-screen data manipulations mode-less and limiting modes to dialogs where user entries must be typed in or where selection between several alternatives must be made. The dialogs are always invoked by a menu command and these are always indicated by an elipsis (...) following the name.

An example of a typical dialog is shown in Figure 10. In the top dialog (Figure 10a), the program is requesting that the user enter some value that it needs to perform the request specified by the user. If the user enters an invalid number, the computer will beep and highlight the offending entry when the OK button is clicked. In the case where multiple requests are made this is a useful feature since the offending entry is made obvious. Should the user continue to enter an invalid entry the program will display an error alert (a special type of dialog) explaining the problem (Figure 10b).

In other dialogs, the user can choose among various options (Figure 11a) or enter preset values (Figure 11b). By only allowing fixed values to be entered using mouse clicks, the possibility of the user entering an invalid entry is eliminated. Note that all the dialogs have one button that is highlighted by a bold outline, this denotes the default action that is taken should a carriage return be entered.

A typical menu is shown in Figure 12. Note that some of the menus and menu items are greyed out, denoting choices that are inappropriate at this time (no data window is open) and are hence, not available. Seeing



(a)

Fourier calibration

Enter sampling increment: 632.8340

OK
Cancel

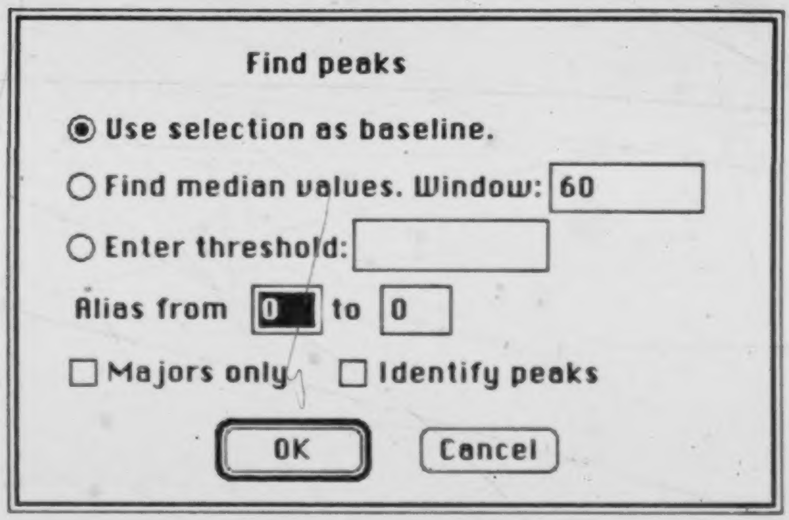
(b)

The value is not a number.

OK
Cancel

Figure 10. An example of a typical dialog (a) and an error alert (b).

(a)



Find peaks

● Use selection as baseline.

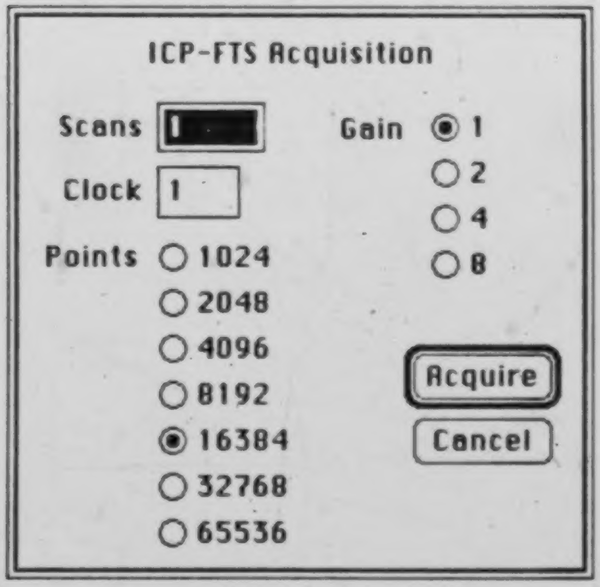
○ Find median values. Window: 60

○ Enter threshold:

Alias from 0 to 0

☐ Majors only ☐ Identify peaks

(b)



ICP-FTS Acquisition

Scans 1

Clock 1

Points ○ 1024 ○ 2048 ○ 4096 ○ 8192 ● 16384 ○ 32768 ○ 65536

Gain ● 1 ○ 2 ○ 4 ○ 8

Figure 11. Some more examples of dialogs.

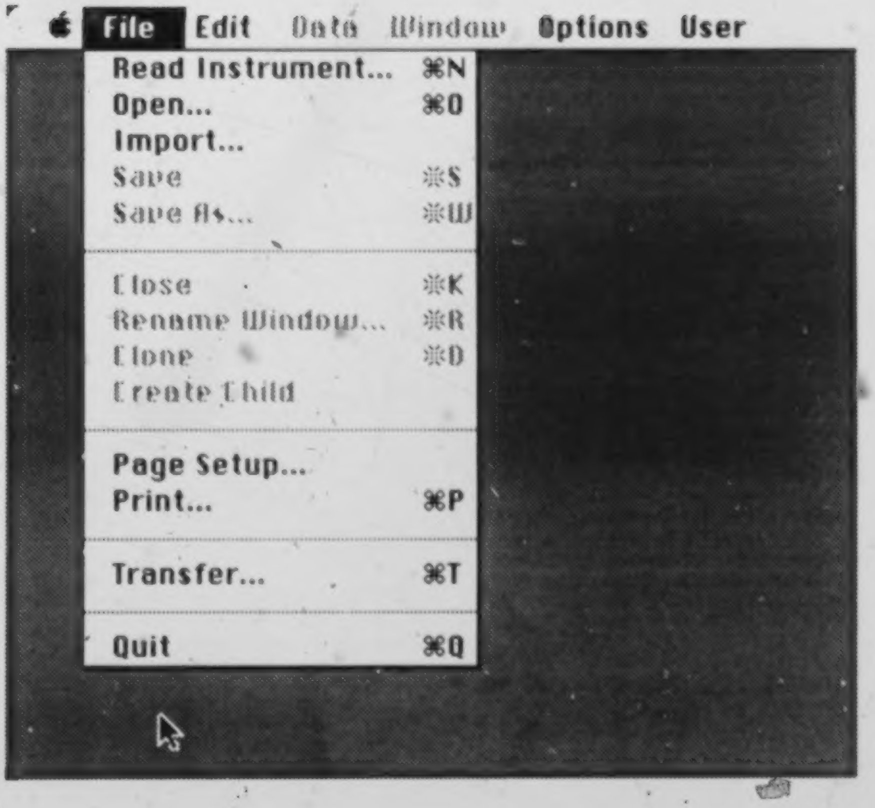

File
Edit
Data
Window
Options
User
Read Instrument... ⌘N
Open... ⌘O
Import...
Save ⌘S
Save As... ⌘W
Close ⌘K
Rename Window... ⌘R
Clone ⌘D
Create Child
Page Setup...
Print... ⌘P
Transfer... ⌘T
Quit ⌘Q

Figure 12. A typical SpectroPlot menu.

which choices are not available is a lot friendlier than making an inappropriate choice and then receiving an error message.

3-3. Handling large data sets

Critical to applications in Fourier transform spectroscopy is the ability to handle large data sets. An average data set from the ICP-FTS instrument in our laboratory is 16,384 (16K) data points and the instrument is capable of acquiring 524,288 (512K) data points at an 80 kHz rate.

Not only must it be possible to have large data sets but operations must be optimized for speed so that response time is minimized in order to preserve interactivity. In addition to optimizing calculation algorithms, screen display and printing algorithms must also be optimized so that the program always appears responsive.

In general, algorithms can be optimized either to conserve memory or for maximum speed, SpectroPlot sacrifices memory for speed. It is for this reason that it was elected to keep all data in memory in IEEE 80 bit extended precision floating point format. Keeping the data in memory eliminates disk access, which is very slow when compared to memory access, and use of extended precision removes the necessity of converting the data (done by the compiler) prior to performing calculations, which are always done using extended precision.

A general rule of thumb in algorithm design is that multiplication, even in hardware, is inherently slow. For this reason, simple additions and binary shifts are substituted for multiplications and divisions whenever possible within loops.

To further keep down computational overhead, the data is always accessed with explicit pointers (memory addresses) instead of array references. Array references generate addresses by adding an offset to the base address of the array. The offset is calculated by multiplying the index by the size of an individual array element. If an array element is referenced several times its address is recalculated each time. This redundant address generation can be eliminated by using pointers.

Further savings can result if the required array elements are equally spaced. The address of the next element can be generated from that of the current element by simply adding a constant. For example, in a loop that accesses consecutive data points each pass through the loop increments a data pointer by the size of a single data point (see Figure 13). This avoids recalculating addresses from scratch which always requires a multiplication to calculate the offset.

In certain parts of the program, performance requirements dictate the use of modules coded in assembly language instead of a high level language like Pascal or C. In SpectroPlot the data acquisition and FFT kernels are written in assembler. The use of assembly language can result in the most efficient code possible since the programmer has full control over all machine resources. High level language compilers, by their machine independent nature, are forced to make assumptions and limit machine access.

The drawbacks to using assembly language are that coding, debugging, and maintenance are more difficult than in high level languages. It is for these reasons that only the performance critical portions of SpectroPlot are

```
TYPE
     xPtr = ^EXTENDED;
VAR
     theData : xPtr; {pointer to data}
     p : xPtr;       {working pointer}
     i : INTEGER;    {loop counter}
     N : INTEGER;    {number of data points}
...
...

p:=theData; {set working pointer to first point}
FOR i:=0 TO N DO
   BEGIN
   p^:=p^/2;  {do something}

   ...        {something else}

   {point to next value}
   p:=POINTER(ORD4(p)+SIZEOF(EXTENDED));
   END;
```

Figure 13. Sample Pascal source code showing direct access of data using pointers and pointer math.

coded in assembler and even then only after they have been prototyped in Pascal.

It is also important to remember that compilers sometimes insert additional code into a program to support debugging. Things like range and overflow checking may take place after each line of source without the programmer being aware of it. In a program that performs many iterations, these additional operations can consume a considerable amount of time. It is therefore important to know what exactly a compiler produces as output code and how to disable time consuming options that are no longer required.

Because of the sheer amount of data that has to be presented to the user, the data is always displayed graphically. Drawing a large number of data points on a screen can be very time consuming because screen access can be rather slow and the process is computationally intensive.

One way to speed up the display is to not draw all the data points if they cannot be resolved, since only one data point per horizontal pixel can be usefully plotted. But it must be remembered that information can be lost in highly compressed graphs.

If multiple data points are plotted in a single column of pixels only the minimum and maximum data points are seen, the rest are obscured. Based on this observation the drawing routine that I have implemented only draws the minimum and maximum values of a group of overlapping data points. This significantly speeds up drawing very large data sets in small windows.

A related problem occurs whenever part of a data window needs to be redrawn because an overlapping window has been closed or moved. The naive approach is to redraw the whole window with the operating system handling any necessary clipping. To save time, the program checks which portion of the window actually needs redrawing and only draws that portion.

An even faster method is to save a copy of the bit image of each window and to directly copy the saved image back whenever an update is required. This method is used extensively in computer games but can consume a lot of memory if there are a lot of windows. In order to maximize the amount of memory available for data, I have elected not to use this method.

The same drawing methods are applied to printing since it suffers from similar problems. Printing routines differ in that they can be interrupted if the user elects to cancel the printing. Print graphics also contain axes and labels which are not drawn on the screen.

The search for ever more speed is not just a goal in itself, speed affects the interactiveness of a program. Slow programs are perceived by users to be less interactive because the user is forced to wait for the computer to catch up, thereby interrupting the user's actions. The goal is to produce a program that the user perceives to be real-time.

3-4. Flexibility

Important for research is the ability to adapt to unpredictable future needs. The ability to alter some function of a program without having to recompile the entire program is essential. This is readily accomplished by writing a program that either has a macro language facility or that can be extended by the user through external modules.

Macro languages have been around for a while in such products as dBase and Lotus 1-2-3. It has allowed these products to penetrate many different markets because they could be modified by the end user to meet different needs. The major problem with macro languages is that they are interpreted and, hence, slow.

The other solution, external modules, is more cumbersome since it requires programming outside the environment of the host application. It more than makes up for this in power and speed since the module is compiled into machine code and has access to all machine resources.

Actually, the optimum solution is to have both options available. The user can prototype new functions with the macro language and once debugged it can be compiled to an external module. An example of a program that implements both these options is HyperCard [71], an application from Apple that has generated much enthusiastic acceptance from Macintosh users and developers.

A serious problem between programmers and researchers is solved by allowing a program to be extended. Researchers frequently demand access to the source code in order to validate the algorithms used by the program but because the source code contains proprietary information programmers commonly withhold it [72]. If the calculation routines are kept in separate modules, then the developer can safely distribute the source to these modules without fear of giving away proprietary information about the main program.

This also simplifies program maintenance because the researcher is now free to modify the calculation routines without affecting the main program. This is very important because it is impossible to maintain a program that exists in hundreds of forms because the source code was distributed and everybody has modified it.

SpectroPlot currently only supports extensions via externally compiled modules. In addition to giving access to extra functionality, some of SpectroPlot's basic functions are implemented as external modules. This allows the user to change these functions by substituting their own modules.

The SpectroPlot data acquisition and file import functions call user supplied modules to perform the actual functions. This allows SpectroPlot to work with any data acquisition hardware and with files from other machines by simply changing modules. In addition, an external module can be supplied that implements a custom x-axis calibration. This allows nonlinear and user specific calibration schemes to be used.

SpectroPlot external modules are implemented as Macintosh standalone code resources, with an entry point at offset zero that appears as a Pascal function (see Figure 14a). The first parameter is a pointer to data for the currently active data window and the second is a pointer to a parameter block in the application's global memory (see Figure 14b) that allows inter-process communications. The function returns an error code to the calling routine. The format of specialized external modules such as external calibration may deviate slightly in that the second parameter may be different.

The parameter block contains the address of a dispatcher routine in SpectroPlot. The called external module uses this address to call SpectroPlot with a request to perform some action. This allows external modules to

(a)

```
FUNCTION xProc(theData: mydataptr;
               xcom: xComPtr): OSErr;
```

(b)

```
xComBlock = RECORD
               dispatcher: ProcPtr;
               request: INTEGER;
               result:  INTEGER;
               inArgs: ARRAY [1..8] OF LONGINT;
               outArgs: ARRAY [1..4] OF LONGINT;
            END;

xComPtr = ^xComBlock;
```

Figure 14. Pascal header for a SpectroPlot external procedure (a) and definition of parameter block (b).

access routines in the SpectroPlot, for example the FFT routine, so that users do not have to write their own.

SpectroPlot also has the ability to read and write ASCII tab delimited data files. This feature allows SpectroPlot to exchange data with commercially available programs that may have features not implemented in SpectroPlot. In addition, the current data visible in a window can be copied onto the clipboard either as text or as a picture, forming another avenue of data exchange.

Being able to export data is very important since SpectroPlot lacks many features found in commercial presentation graphics programs. An example of what can be done using commercially available software is shown in Figure 15. Here printed output from SpectroPlot was captured in a file using a special printer driver and then transferred to a graphics program where the arrows and annotations were added. The entire figure was then scaled so that it could fit within the margins of a thesis page. The various commercial programs used were (in order of use), Glue, MacDraw II, and Word.

Data can also be exported to commercial spreadsheet and statistical programs for further analysis [67]. This is desirable because it doesn't require that the user write an external module to perform a simple "one-of" calculation. This enhances creativity in that it makes it easier to process one's data in non-standard ways.

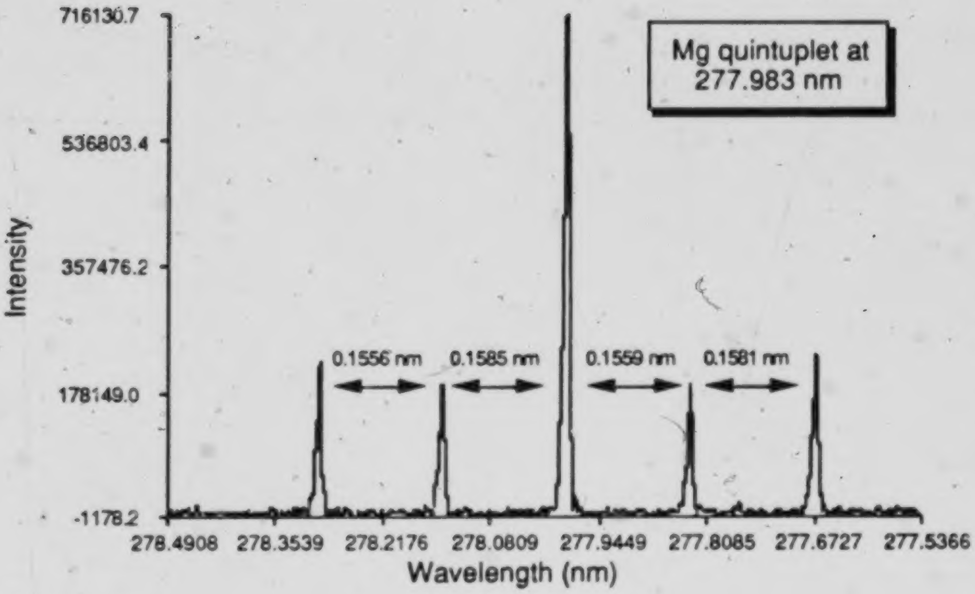


Figure 15. An example of SpectroPlot output that has been modified using a graphics program.

3-5. Programming paradigm

Graphical user interfaces are more complex to program than traditional text interfaces. In order for them to be accepted, programmers must first be persuaded to write them. Programmers tend to resist writing graphical interfaces because they perceive few benefits in return for the amount of effort expended. It typically requires between six months and a year for a programmer to learn how to implement a desktop interface in a major application that conforms to commercial standards of quality.

In order to facilitate development and implementation of advanced graphical interfaces, a new programming paradigm is required to replace the procedural style currently used by most programmers. A paradigm shift is required because the new user interfaces are based on non-procedural models that can be difficult to represent in procedural code. For programmers to shift to a new paradigm, it must offer clear benefits that offset the time and effort required to make the shift.

Such a new paradigm is object oriented programming (OOP), OOP is a logical extension of structured programming. The structured programming themes of code and data abstraction are expanded upon to form the core of OOP. In structured programming, the programmer was free to use whatever level of abstraction (s)he desired but OOP enforces abstraction. Both code and data in OOP are combined into objects and actions are carried out through messages to the objects.

OOP is particularly suited for implementing graphical interfaces because the code can reflect the structure of the graphical objects on the screen. Windows, for example, can be one type of object. In addition to containing the description of the window and the window data, the object also contains the routines required to draw the window and handle user interaction. The window object knows how to draw itself on the screen. Instead of calling a routine to draw a particular window, the programmer simply sends a message to the target window to draw itself.

Though OOP has been used extensively in the academic environment, it is just starting to become a commercially accepted practice. In academia, interpreters such as Lisp and Smalltalk are favoured while commercial programmers had to wait for C++ and Object Pascal compilers. A detailed description of OOP is beyond the scope of this thesis and the interested reader is referred to Schmucker's text for the Macintosh environment [73].

OOP code is easy to produce but is often less efficient than old fashioned procedural code. It is possible to use OOP techniques with existing procedural languages [74] and this was the method used to write SpectroPlot. This was a compromise between procedural and OOP methods that allowed the best features of both to be combined. The Pascal compiler used to produce SpectroPlot has OOP features but none were used.

3-6. Conclusions

The goal of all this effort is to produce a program using the latest advances in software design that shows how software can enhance the researcher's ability to interpret her/his data. True interaction is stressed rather than just simple automation to produce productivity gains. In order to make the most use of computer resources, the program is integrated into the overall software environment.

Routine usage in our laboratory has shown that SpectroPlot is easy to use, though its multitude of options for advanced users still requires a cleaner interface. SpectroPlot's expandability has allowed new experiments and data reduction methods to be investigated with minimal new software requirements. Combined with existing commercial software it has become easier to produce presentation graphics from our data, in fact manual paste-up has been eliminated and camera ready copy is produced directly from the computer.

Software has become the limiting factor in achieving the full potential of computers. Researchers must realize that the software environment is as important as the hardware and must be factored into the cost and design of the system. Future systems will continue the trend towards increasing importance of software as artificial intelligence techniques become integrated into main-stream applications.

Chapter 4

Qualitative atomic spectrochemical analysis.

Fourier transform spectrometers are superior to conventional scanning monochromator based systems in that they acquire an entire spectrum simultaneously which can result in enormous time savings. In addition, FTS instruments offer absolute wavelength accuracy and high resolution in a compact package. These factors make FTS instruments especially attractive for qualitative atomic spectrochemical analysis (i.e. elemental composition identification) and the compilation of spectral atlases, a tedious task.

In this chapter, the qualitative performance of an ICP-FTS system is evaluated. In order for qualitative uv-vis data acquired on FTS instruments to be accepted, it must be shown that the wavelength axis is indeed accurate and that the technique introduces no spurious spectral lines into the spectrum. These are the two major criteria to be assessed in evaluating our system. In addition, the resolution of the system is also briefly discussed since it is an important criteria for a "line rich" emission source such as the ICP.

4-1. Aliasing

Aliasing is the name given to the phenomenon that is observed when the Nyquist criterion isn't satisfied when sampling (digitizing) a continuous function. Nyquist states that in order to properly sample a waveform with a maximum frequency f, the sampling frequency must be 2 × f. When under-sampling occurs the frequency axis folds such that each point can appear to have more than one frequency and these additional frequencies are called aliases. Aliasing introduces an element of ambiguity into the sampled waveform and is normally avoided by limiting the bandwidth of the input waveform to one half the sampling frequency.

Our interferometer system is capable of acquiring data over a mirror movement equivalent to ±2 cm of optical retardation on either side of the zero path difference (ZPD) point. This maximum path difference corresponds to 65,536 HeNe laser clock cycles. The actual data acquisition clock frequency can be one of 1, 4, or 8 times that of the basic HeNe clock, and the various aliasing regions for these clock frequencies are shown in Table 2. In order to properly sample the entire spectral range down to 150 nm, the 8× clock must be used because the other two clocks would alias the resulting spectrum. Therefore, to properly sample the entire uv-vis region at maximum resolution 8 × 65,536 (or 524,288) data points must be acquired.

The computer currently used with our interferometer system is a Macintosh II from Apple Computer. In its maximum configuration using the Macintosh OS, 8 megabytes of RAM can be supported. With the data represented as 80 bit IEEE extended precision floating point numbers, this is sufficient memory to transform 262,144 data points in memory. We actually only have 2 MBytes of memory installed so we are currently limited to 65,536 data points. This only allows us to properly sample interferograms (i.e. without aliasing) for 8192 laser cycles, which corresponds to only one eighth of the maximum resolution that can be attained by our instrument.

The solution to this data handling problem is to undersample and accept aliased spectra. Aliased spectra resemble those acquired from an

Aliasing region	Clock (× HeNe) 1	4	8
0	0 - 7901 cm^{-1} (∞ - 1266 nm)	0 - 31,605 cm^{-1} (∞ - 316 nm)	0 - 63,209 cm^{-1} (∞ - 158 nm)
1	15,802 - 7901 cm^{-1} (633 - 1266 nm)	63,209 - 31,605 cm^{-1} (158 - 316 nm)	
2	15,802 - 23,704 cm^{-1} (633 - 422 nm)		
3	31,605 - 23,704 cm^{-1} (316 - 422 nm)		
4	31,605 - 39,506 cm^{-1} (316 - 253 nm)		
5	47,407 - 39,506 cm^{-1} (211 - 253 nm)		
6	47,407 - 55,308 cm^{-1} (211 - 181 nm)		
7	63,209 - 55,308 cm^{-1} (158 - 181 nm)		

Table 2. Aliasing regions in both wavenumbers and nanometers for sample clocks of 1, 4, and 8 times the basic HeNe clock frequency.

echelle spectrometer without an order sorter, all the various orders being superimposed on top of each other. Each spectral element can therefore have more than one wavelength, or alias, assigned to it.

An aliased Mg hollow cathode lamp (HCL) spectrum is shown in Figure 16. An R166 solar blind PMT, which has a wavelength response range of approximately 180 nm to 310 nm, was used as a detector. The three wavelength axes shown for the spectrum correspond to the aliasing regions that lie within the bandpass of the detector.

It is possible to tolerate aliasing in atomic emission spectroscopy because the emission lines are so well characterized. The only problem is that the probability of spectral overlap in complex spectra is increased. Our research group has used aliasing extensively in the past to maximize resolution when the amount of data that can be handled is limited [10]. Recently, Wang and Marshall [75] have utilized extreme (1000 fold) aliasing to acquire high resolution FT/ICR spectra with only 2048 data points. This points to another advantage of aliasing, the short data sequences can be transformed in real-time.

By bandpass limiting either the source or the interferogram, it is possible to "de-alias" an undersampled spectrum. This in effect eliminates some of the possible wavelength assignments for each spectral point since they will not be in the bandpass. A fully de-aliased spectrum would have unique wavelength assignments just like a properly sampled spectrum.

An example of a de-aliased spectrum is the acquisition, with a 4× clock, of the ultraviolet region using a solar blind PMT as the detector. This spectrum would normally have two wavelengths assigned to each spectral point

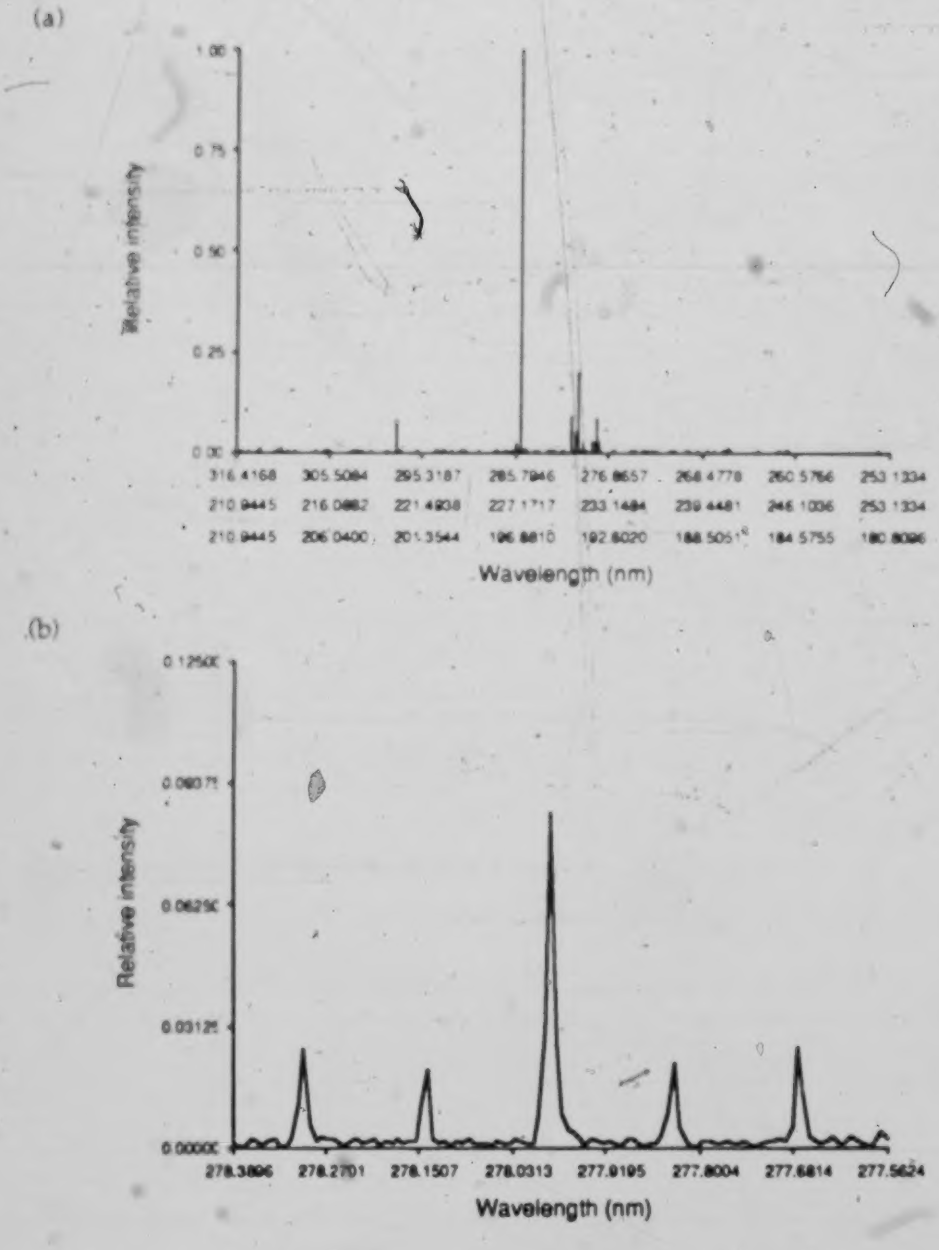


Figure 16. Aliased Mg HCL spectrum showing the wavelength regions when an R166 PMT is used as a detector (a) and a close-up of the quintuplet at 277.983 nm (b).

but since the solar blind PMT doesn't respond to visible radiation these wavelengths do not have to be considered. An aliased spectrum of Fe is shown in Figure 17a along with a de-aliased spectrum (Figure 17b). There are four possible wavelength axes for the aliased spectrum that covers the same spectral region as the de-aliased spectrum.

Aliasing allows us to maximize the spectral resolution achievable on our instrument without incurring debilitating data overhead costs. The penalty for this performance is more complex spectra and multiple wavelength axes, due to spectral folding. When unambiguous wavelength assignment is required, de-aliasing can be used in lieu of sampling at the proper Nyquist rate.

4-2. Wavelength calibration

In Fourier transform spectrometers, the wavelength axis of the spectrum is a function of the intervals at which the interferogram is sampled and the total number of samples. This is an absolute relationship and because the interferometer is laser referenced this forms the basis for the inherent accuracy of FTS acquired wavelengths. In actual practice, high resolution FTS systems do require wavelength calibration with wavelength standards [76].

The actual equation for calculating the wavenumber position, v_i, of a specified spectral point (the i'th point) from the wavelength of the HeNe reference laser (λ_{HeNe}) is:

$$v_i = \frac{i}{N \lambda_{HeNe}}$$ where N = the total number of points
and $i = 0, 1, 2, \ldots, N-1$ Equation 4-1

The wavelength in nanometers is then derived from the wavenumber value.

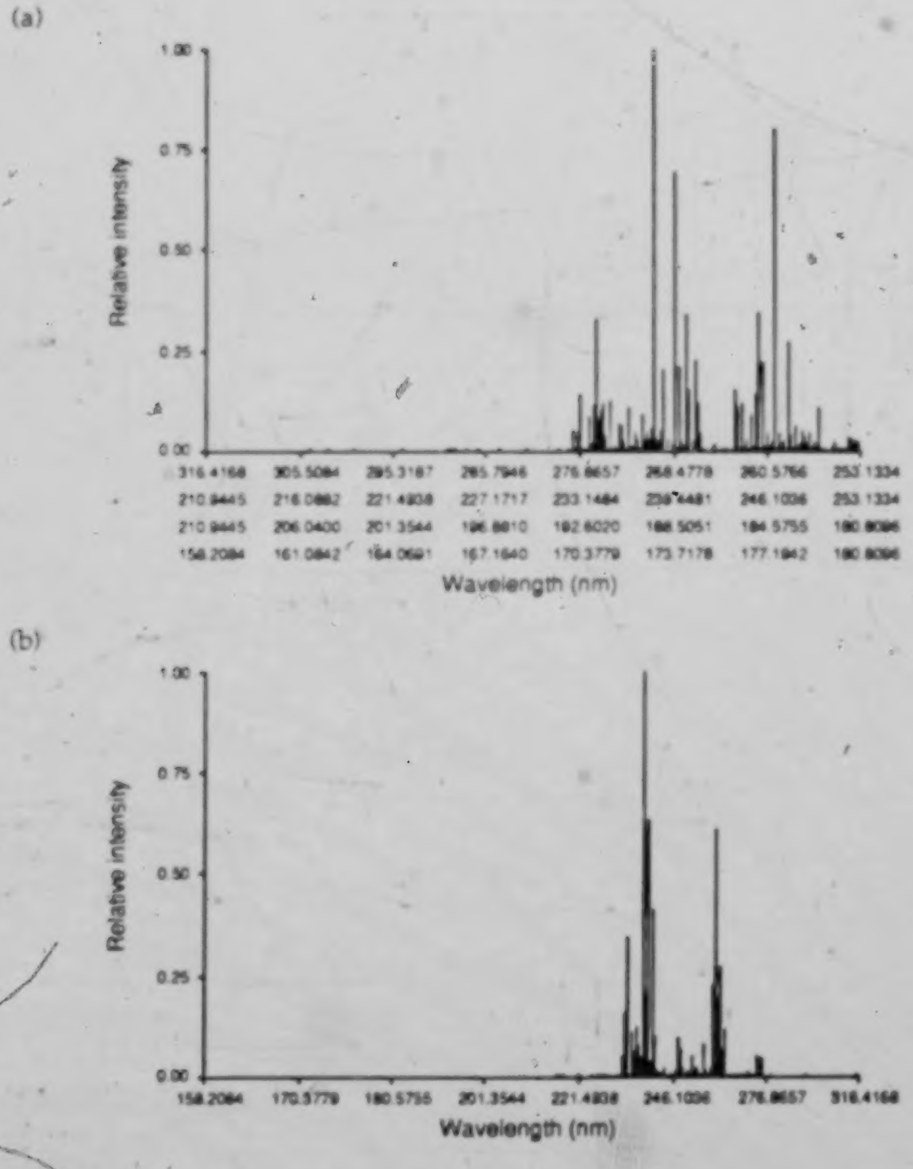


Figure 17. Aliased Fe ICP spectrum (a) showing the wavelength regions corresponding to the de-aliased spectrum (b).

In our interferometer system, wavelengths calculated on the basis of the standard 632.81646 nm wavelength of the HeNe reference laser [77] have been found to be slightly in error (see Table 3). The Mg I quintuplet at 277.983 nm in the Mg HCL spectrum (Figure 16b) was used for this calculation. The spectrum was calculated from an aliased 64K interferogram. Note that the error is determinate. The possible reasons for the discrepancy are varied (use of non-mode-locked HeNe laser, mis-alignment of reference channel optics, and optical differences between reference and sample channels) but the result is that the wavelength axis of resulting spectra must be calibrated.

Wavelength calibration of our system was implemented using NBS Fe wavelengths [78] as standards, and the lines used are shown in Figure 18b. A novel calibration scheme was used whereby Equation 4-1 was used with the measured position of a reference line and the NBS wavelength value to calculated a new wavelength value for the HeNe reference laser. The average value of the "calculated" HeNe wavelengths was then used to calculate future wavelengths in sample spectra.

After calibration, the wavelength values obtained from our instrument are in better agreement with literature values (see Table 4). The remaining discrepancies are on the order of the uncertainty of the literature values (±0.001 nm). Also note that our calibration scheme resulted in correct wavelength values at 278 nm though the wavelength axis was calibrated at 240 nm. The minor systematic discrepancy may be due to refractive index differences.

Because the spectrum is digitized at discrete intervals, it is possible for a line center to not coincide with a spectral point. The discrete nature of the resulting spectrum limits the accuracy of measuring the wavelength of a line

Literature λ (nm)	Observed λ (nm)	Error (nm)
278.297	278.2888	-0.0082
278.142	278.1334	-0.0086
277.983	277.9749	-0.0081
277.827	277.8185	-0.0085
277.669	277.6609	-0.0081

Table 3. Observed wavelength values for the Mg quintuplet when the wavelength axis is calibrated using the reference value for the HeNe wavelength.

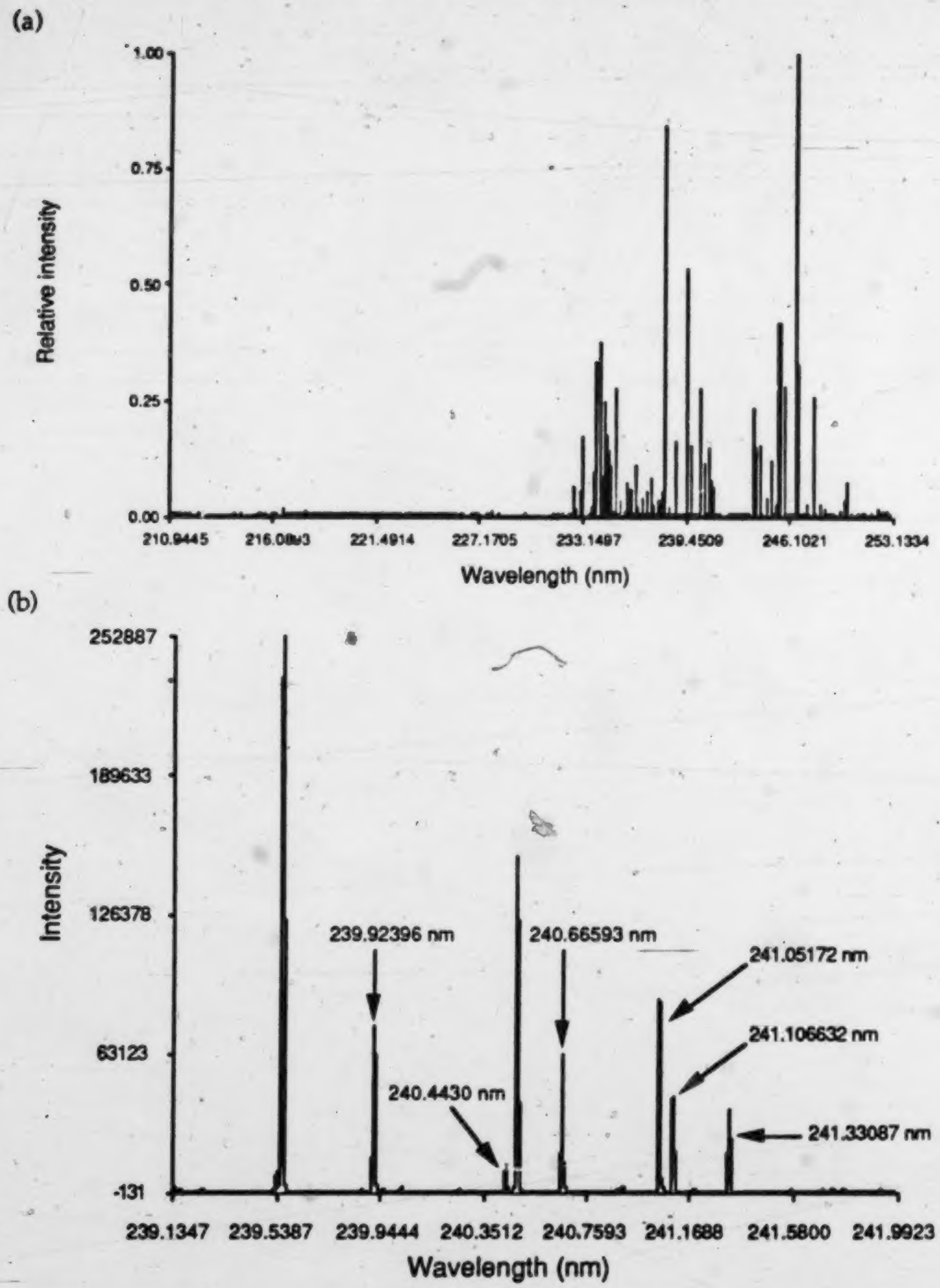


Figure 18. Aliased Fe ICP spectrum (a) and a portion of the Fe spectrum showing the NBS wavelengths used to calibrate the wavelength axis (b).

Literature λ (nm)	Observed λ (nm)	Error (nm)
278.297	278.2960	-0.0010
278.142	278.1404	-0.0016
277.983	277.9819	-0.0011
277.827	277.8260	-0.0010
277.669	277.6679	-0.0011

Table 4. Observed wavelength values for the Mg quintuplet when the wavelength axis is calibrated using a corrected value for the HeNe wavelength calculated from NBS Fe reference lines.

to ± ½ point. The consequences of this digitization error, for two Mg lines are shown in Figure 19.

In order to obtain more accurate wavelength values, the experimental spectra that were used to calibrate the wavelength axis and test wavelength accuracy were zero filled four times in order to obtain a better estimate of the position of the peak maxima. Zero filling refers to extending the interferogram to a longer length with zeros before transforming. Zero filling four times means the interferogram is extended to four times its original length.

In our current data system, we do not have sufficient memory to do this to an entire 64K interferogram. Instead, a section of the spectrum is inverse transformed to produce a new shorter interferogram which is then zero filled and transformed back to produce an interpolated spectrum of the section. A four times zero filled spectrum from a 64K interferogram sampled at the laser clock rate would have a wavelength measurement resolution of 0.06125 cm^{-1} which corresponds to 0.0005 nm at 278 nm.

This type of digitization error occurs for all spectroscopic techniques that store spectra as discrete points and is well recognized, for example in photodiode array spectrometer systems [79]. For this reason, whenever maximum accuracy is required the spectrum is zero-filled to interpolate the lineshape so that the peak maxima can be located more precisely. In addition, interpolation yields more accurate intensity values.

Lines with wavelengths ≤ 200 nm are usually measured with vacuum spectrometers. Our instrument is sufficiently sensitive to be able to observe lines in the region from 190 to 200 nm in air, but the difficulty is that the liter-

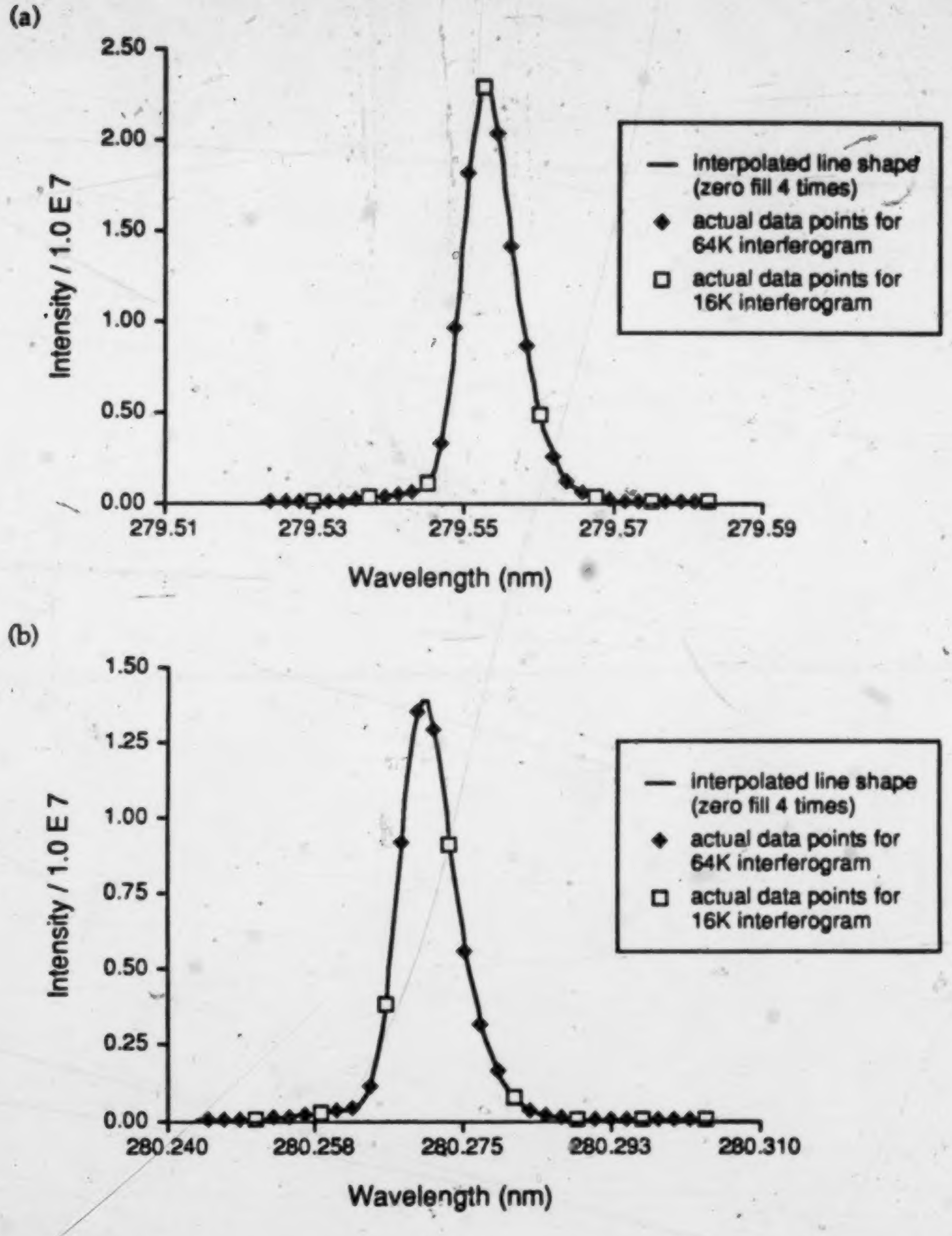


Figure 19. Actual spectral data and interpolated lineshapes for Mg 279 (a) and 280 (b) nm emission lines.

ature wavelengths are for vacuum. Because the region is narrow, it was possible to derive a linear relationship between our measured air wavelengths and the corresponding literature vacuum values.

Various HCL and ICP lines in this region (see Table 5) were used to calculate a linear fit and good agreement between the linear relation and the actual values was obtained (see Figure 20). The resulting linear equation is given in Equation 4-2 and all units are in nm.

$$\lambda_{air} = 0.999663 \times \lambda_{vac} + 0.001385 \qquad \text{Equation 4-2}$$

The correlation coefficient for this linear fit is 0.99999999. Currently this calibration is done after the fact but it is possible to incorporate it into the wavelength readout routines of our spectral analysis software.

4-3. Resolution

Because ICP emission spectra can have a high spectral density, it is important that the spectrometer have high resolution in order to minimize spectral overlaps. Our current interferometer has a maximum nominal resolution of 0.5 cm^{-1} based on its maximum optical retardation of 2 cm. This corresponds to 2 pm at 200 nm, giving a theoretical resolving power of 100,000. This kind of resolving power approaches that of high resolution echelle spectrometers [80]. It should be noted that the resolution is constant in wavenumbers. The corresponding wavelength resolutions for a 1 cm^{-1} wavenumber resolution at various wavelengths are shown in Table 6. The actual spectral resolution would depend upon the type of apodization that is applied to the interferogram.

Element	State	Vacuum λ (nm)	Air λ (nm)
Ne	I	190.749	190.686
Ne	I	191.608	191.545
Ne	I	193.003	192.939
Ne	I	193.883	193.819
C	I	193.091	193.028
Cu	II	200.035	199.969

Table 5. Lines used to calibrate the wavelength region between 190 nm and 200 nm.

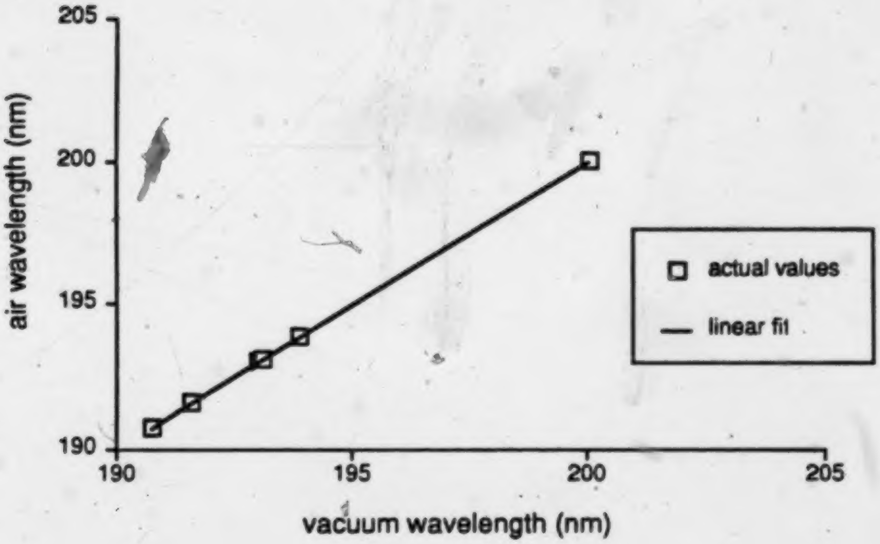


Figure 20. Air-vacuum wavelength calibration curve.

Wavelength (nm)	Resolution (nm)
200	0.00400
250	0.00625
300	0.00900
350	0.01225
400	0.01600

Table 6. Calculated wavelength resolutions at various wavelengths for a 1.0 cm^{-1} wavenumber resolution.

The actual spectral resolution that can be achieved by our instrument can be measured using a very narrow line source. The Zn 213.8 nm emission from a hollow cathode lamp as measured on our instrument is shown in Figure 21. This line has a Doppler width of about 0.33 pm [81]. The spectrum was calculated from a 64K interferogram sampled with the basic laser clock and the region around 213.8 nm was zero filled four times to interpolate the line shape for more accurate measurement. The interferogram decayed to zero within the ±2 cm of optical retardation that it measured over, so no additional apodization was applied.

The observed full width at half maximum is 1.04 cm^{-1} which corresponds to approximately 4.75 pm at 213.8 nm. Since the physical linewidth is much smaller than this, it is a good estimate of the instrument line width. Since ICP emission lines have FWHM values typically around 2 to 5 pm, our instrument is not quite capable of fully resolving ICP line profiles. However, Boumans [82] has determined that the optimum spectrometer resolution for practical analytical work is on the order of the physical linewidth of the lines being measured.

Another illustration of the resolution achieved by our instrument is seen in Figure 22 which shows a 0.5 nm portion of an Fe ICP spectrum calculated from a 64K interferogram sampled with the basic laser clock. Two sets of wavelength axes are shown because the spectrum is aliased. In this spectrum, the 234.810/234.830 nm lines are almost baseline resolved. For the ICP, lines closer than 0.010 nm apart usually cannot be separated regardless of spectrometer resolution because of line broadening [83].

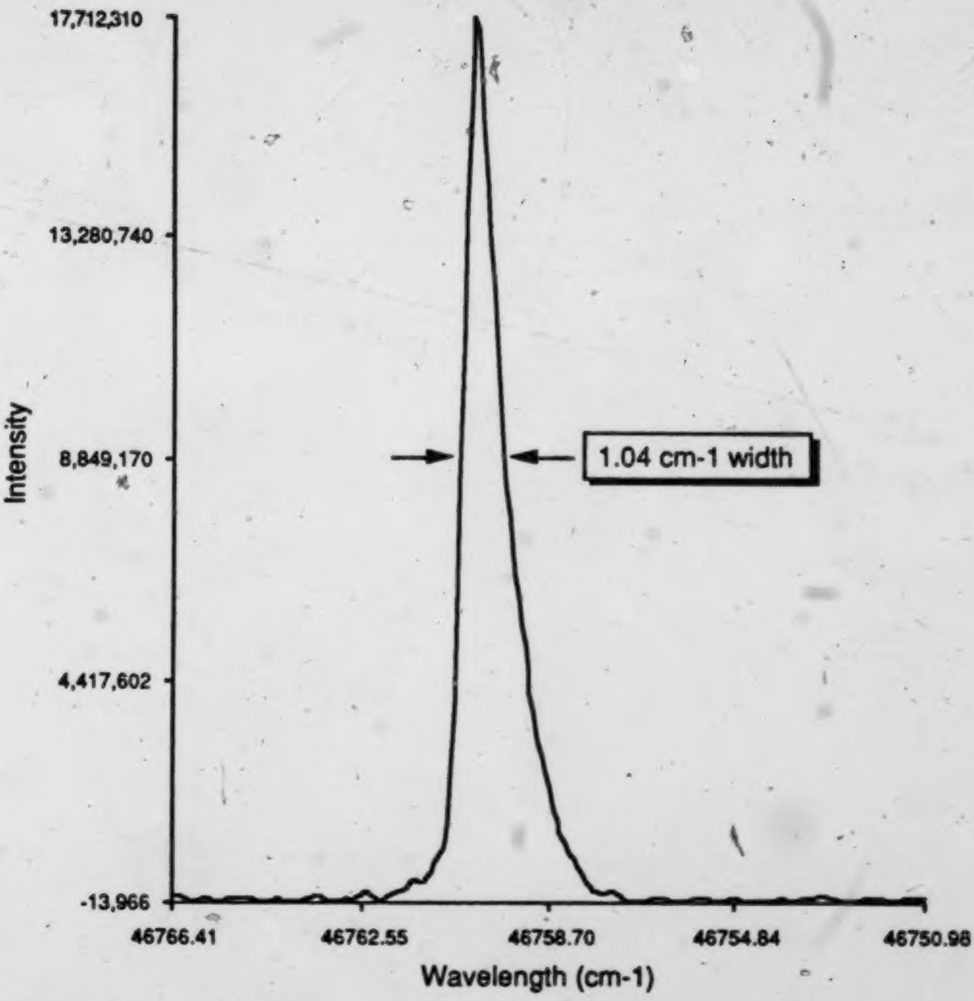


Figure 21. Hollow cathode emission of the Zn 213 nm line, 64K interferogram.

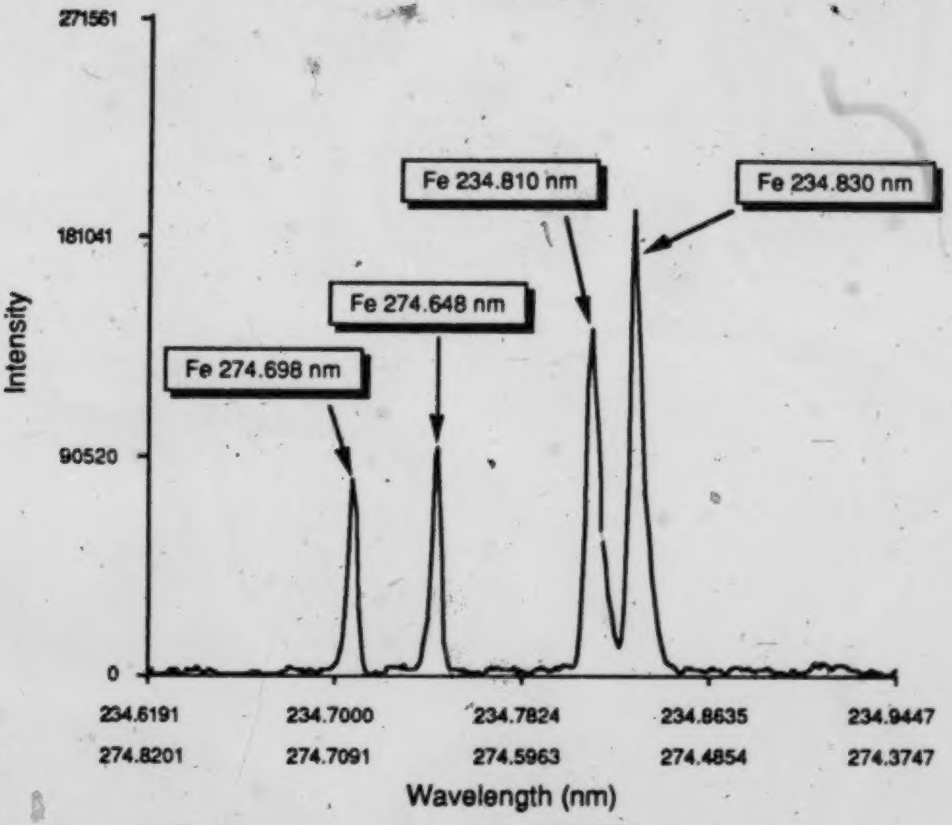


Figure 22. An aliased 0.5 nm portion of an ICP spectrum of Fe calculated from a 64K interferogram. 1000 mg/L Fe.

4-4. Spurious spectral lines

Spurious spectral lines in an FTS acquired spectrum can arise from two sources, modulation of the emission source and interferogram sampling rate errors. Both are related since sampling rate errors can in effect modulate the source.

Starting with a single spectral line whose intensity is being modulated in a simple sinusoidal fashion, the resulting interferogram can be calculated and the nature of the spurious spectral lines understood using Fourier analysis. The following derivation is for an ideal interferometer that is perfectly aligned and coupled to the optical source.

The relationship between the measured interferogram $I'(\delta)$ and the measured spectrum $B'(\nu)$ is shown in Equation 4-3.

$$I'(\delta) = \int_{-\infty}^{\infty} B'(\nu)\, e^{-i2\pi\delta\nu}\, d\nu \qquad \text{Equation 4-3}$$

The measured spectrum can be thought of being composed of the product of the generalized ideal spectrum $B(\nu)$ and an amplitude function $A(t)$. Substituting into Equation 4-3 gives:

$$I'(\delta) = \int_{-\infty}^{\infty} A(t)\, B(\nu)\, e^{-i2\pi\delta\nu}\, d\nu \qquad \text{Equation 4-4}$$

and rearranging gives:

$$I'(\delta) = A(t) \int_{-\infty}^{\infty} B(\nu)\, e^{-i2\pi\delta\nu}\, d\nu \qquad \text{Equation 4-5}$$

The ideal interferogram $I(\delta)$ of the ideal spectrum $B(\nu)$ is given by

$$I(\delta) = \int_{-\infty}^{\infty} B(v)\, e^{-i2\pi\delta v}\, dv \qquad \text{Equation 4-6}$$

and can be substituted into Equation 4-5 to give:

$$I'(\delta) = A(t)\, I(\delta) \qquad \text{Equation 4-7}$$

If the spectrum B(v) is composed of a single infinitely narrow line, v_0, then

$$I(\delta) = 1 + \cos(2\pi\delta v_0) \qquad \text{Equation 4-8}$$

which is substituted into Equation 4-7 to give:

$$I'(\delta) = A(t) + A(t)\cos(2\pi\delta v_0) \qquad \text{Equation 4-9}$$

The amplitude function A(t) is composed of the sum of various modulation components. In the simplest case, the amplitude of the spectrum is modulated in a sinusoidal fashion at a single frequency A_m.

$$A(t) = A_0 + A_m \cos(2\pi f t) \qquad \text{Equation 4-10}$$

A_0 is the unmodulated spectral intensity of the spectral source.

The relationship between optical retardation δ and time t depends upon the velocity of the moving mirror.

$$\delta = 2\bar{v}t \qquad \text{Equation 4-11}$$

Where $\bar{v}$ is the velocity of the mirror.

$$\bar{v} = \frac{x}{t} \qquad \text{Equation 4-12}$$

Using this relationship, the amplitude function can be expressed in terms of optical retardation.

$$A(\delta) = A_0 + A_m \cos\left(\frac{\pi f \delta}{\bar{v}}\right) \qquad \text{Equation 4-13}$$

This new amplitude function is used with Equation 4-9 to give:

$$\Gamma(\delta) = A_0 + A_m \cos\left(\frac{\pi f \delta}{\bar{v}}\right) + A_0 \cos(2\pi\delta v_0) + A_m \cos\left(\frac{\pi f \delta}{\bar{v}}\right) \cos(2\pi\delta v_0) \qquad \text{Equation 4-14}$$

By using the trigonometric function

$$\cos(\alpha)\cos(\beta) = \frac{1}{2}\cos(\alpha - \beta) + \frac{1}{2}\cos(\alpha + \beta) \qquad \text{Equation 4-15}$$

and making $\alpha = 2\pi\delta v_0$ and $\beta = \frac{\pi f \delta}{\bar{v}}$ the following description of the measured interferogram is obtained.

$$\Gamma(\delta) = A_0 + A_m \cos\left(\frac{\pi f \delta}{\bar{v}}\right) + A_0 \cos(2\pi\delta v_0) + \frac{A_m}{2} \cos\left(2\pi\delta v_0 \pm \frac{\pi f \delta}{\bar{v}}\right) \qquad \text{Equation 4-16}$$

From Equation 4-16 it is seen that the resulting interferogram is composed of five frequencies, the DC term, the fundamental frequency produced by the spectral line, the modulation frequency, and two equally spaced sidebands resulting from the modulation of the spectral line. Under normal conditions, the modulation frequency and the DC terms are removed by high pass filtering the interferogram. Transforming this interferogram will result in a spectrum with three peaks, the spectral line surrounded by the two sidebands.

Such a result is consistent with double side band modulation of a carrier wave [84], the carrier wave being the interferogram of v_0. This was used by Marra and Horlick [21] to explain the presence of small satellite peaks in their spectra. Sing and Hubert [85] also use a similar model as a basis for their intensity ratioing technique.

This model can be tested by measuring the spectrum of a modulated source using a Fourier transform spectrometer. We have done so using a Mg hollow cathode lamp as a source and physically chopping the beam with a mechanical chopper (Rofin Ltd. Frequency Programmable Light Chopper, Mk. II). The physical set-up is shown in Figure 23. An R166 solar blind PMT was used as the detector so that only the ultraviolet spectrum was recorded. The interferogram was sampled using the basic HeNe reference clock and hence, the spectrum is aliased. At the time this experiment was performed the velocity of the mirror was ~0.15 cm/s which resulted in a HeNe clock frequency of 5140 Hz. The interferograms are 16,384 points long and are the sum of eight scans.

Because the Mg HCL emission spectrum (Figure 24a) has several lines in the uv, each line is expected to have its own set of sidebands. Fortunately, a single line (285.213 nm) dominates the spectrum. In addition, because a mechanical chopper is used the modulation waveform is a square wave which will result in sidebands at all the odd harmonics of the chopping frequency, further complicating the observed spectrum.

Part of a modulated Mg spectrum is shown in Figure 24b. Note the presence of two new strong lines. These are the modulation sidebands of the Mg 285 nm line. This conclusion is confirmed by the symmetry of the new

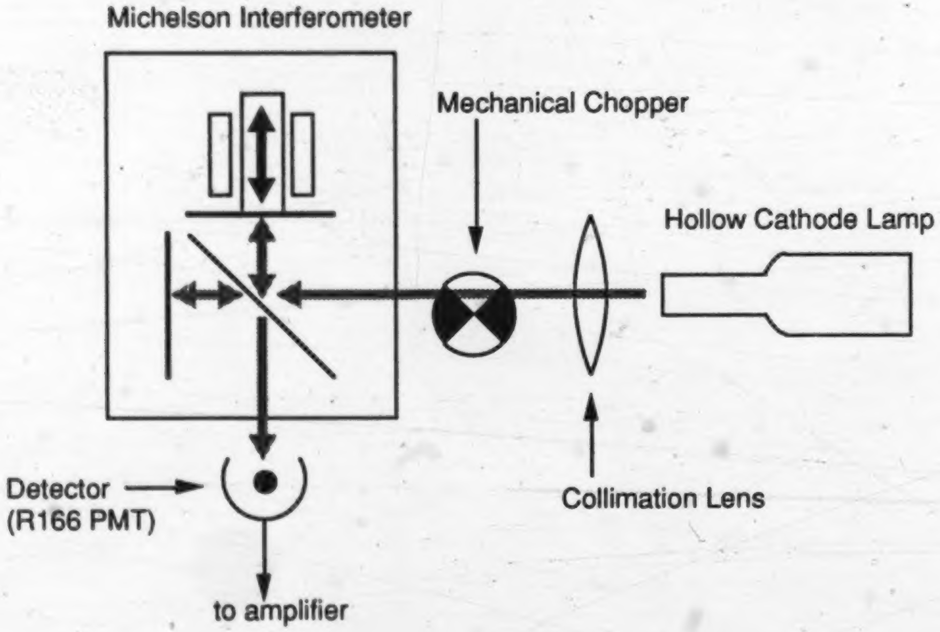

Michelson Interferometer
Mechanical Chopper
Hollow Cathode Lamp
Detector
(R166 PMT)
Collimation Lens
to amplifier

Figure 23. Physical set-up for modulation experiment.

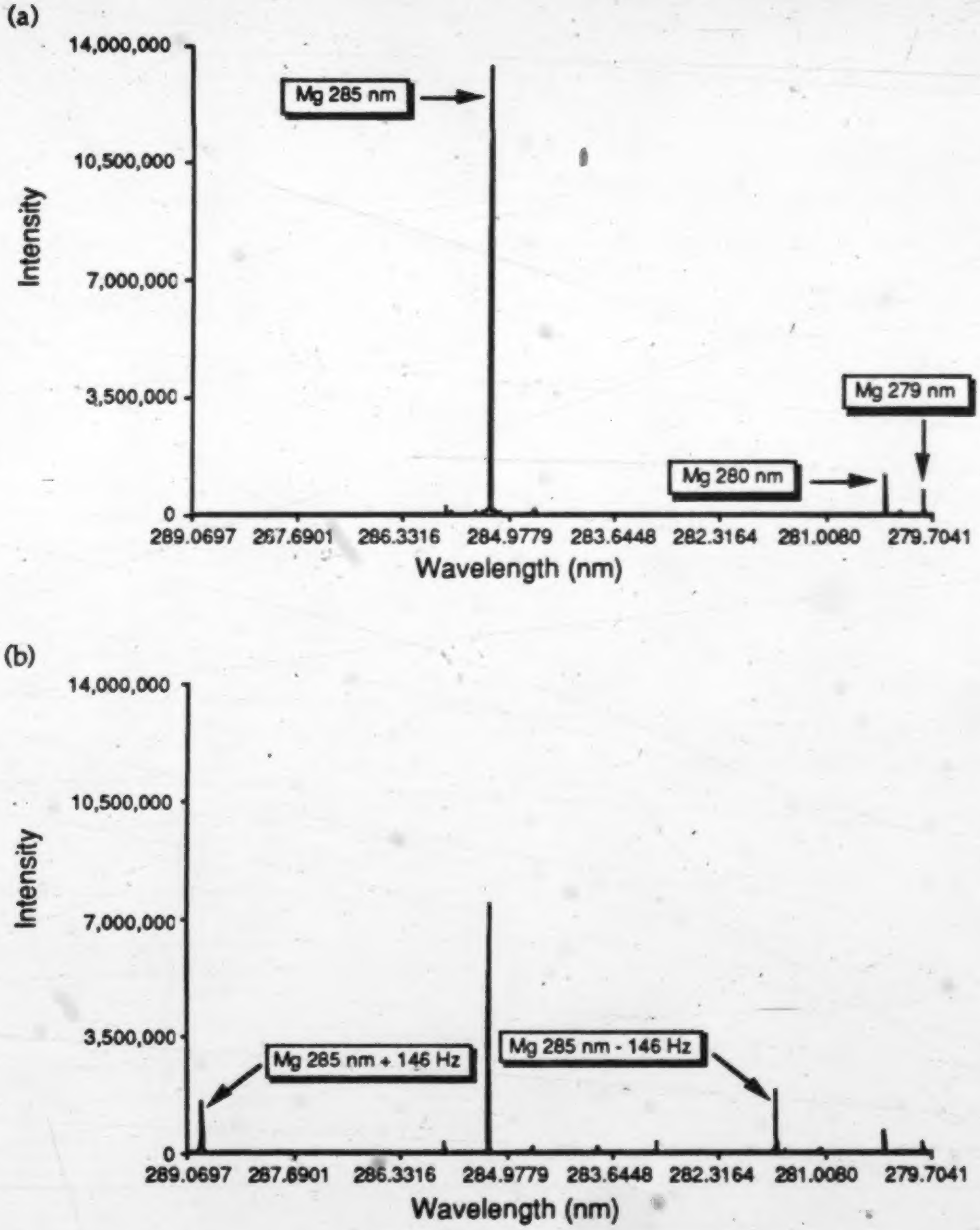


Figure 24. Unmodulated (a) and 146 Hz modulated (b) Mg hollow cathode lamp spectra.

lines about the source line and the dependence of their separation on the modulation frequency.

Also note that the spectral intensities in the modulated spectrum are lower than those of the un-modulated spectrum. This is due to the loss of light when the beam is off. Because FTS is an integrating technique, the measured intensity is the sum of the intensity over the length of the interferogram.

The results of varying the modulation frequency are summarized in Figure 25, which is a plot of modulation frequency vs. sideband separation from the source line. The modulation frequency was measured by observing the chopper sync output on an oscilloscope. The calculated frequencies were obtained from the sideband positions in the spectra based upon a 5140 Hz HeNe clock rate.

Because the moving mirror in the interferometer is traveling at constant velocity, the sampling of the interferogram by the ADC can be considered a function of time as well as optical path difference. This makes the wavelength axis of the resulting spectrum also a frequency axis. Each point in the spectrum corresponds to 0.314 Hz/point (5140 Hz/16384 points).

Multiplying the above value by the separation of the satellite from the source line gives the calculated modulation frequency. These values are summarized in Table 7. The agreement between the measured and calculated frequencies is quite good (~3%). The difference between the two sets of frequencies is most likely due to errors in reading the oscilloscope traces of the sync output.

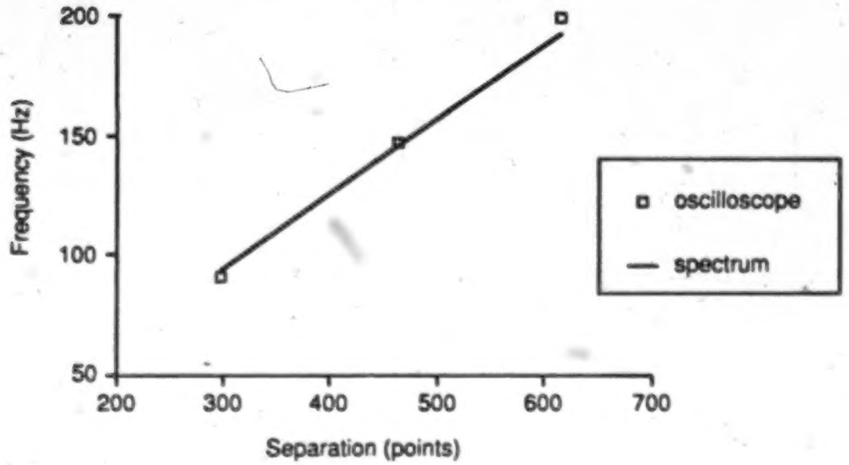


Figure 25. Plot of sideband frequency as a function of separation showing both calculated and observed frequencies.

Sideband separation (# of points)	Measured frequency (Hz)	Calculated frequency (Hz)
298	90.91	93.49
465	147.06	145.88
616	200.00	193.25

Table 7. Measured and calculated modulation frequencies from HCL modulation experiment.

The accuracy of the calculated values is supported by data shown in Figure 26b which shows a Mg ICP emission spectrum in the vicinity of the 279 nm and 280 nm lines. This particular spectrum was chosen because our early ICP spectra [21] were prone to powerline interference. Improved isolation has since eliminated or reduced this interference. Sidebands due to power line modulation are present for both the Mg emission lines and their calculated frequencies are 59.9 Hz and 119.8 Hz with a digital resolution of ±0.3 Hz. These values agree very well with the multiples of 60 Hz associated with power line interference.

The 23 Hz modulation sidebands seen in Figure 26 are seen in all spectra and are the result of a small ripple in the drive control circuitry. This causes a small modulation of the interferometer mirror drive which in turn modulates the interferogram.

Because the modulation that causes the various sidebands is not phase coherent with respect to the interferogram, signal averaging reduces the intensity of the sidebands. The decrease in the relative intensity of the 23 Hz sidebands with respect to its parent peak as the number of signal averaged scans increases is shown in Figure 27. The observed decay matches the $1/\sqrt{\text{scans}}$ relationship expected for the signal averaging of random noise.

Further away from the parent peaks are a series of broad modulation features that only appear in ICP spectra. These features are located at approximately 300 Hz on either side of the parent which roughly corresponds to modulation features found in the ICP noise power spectrum [86]. These plasma modulation bands are shown in a Mg spectrum (Figure 28). Note that this is not the same spectrum as shown in Figure 26. Increased physical isola-

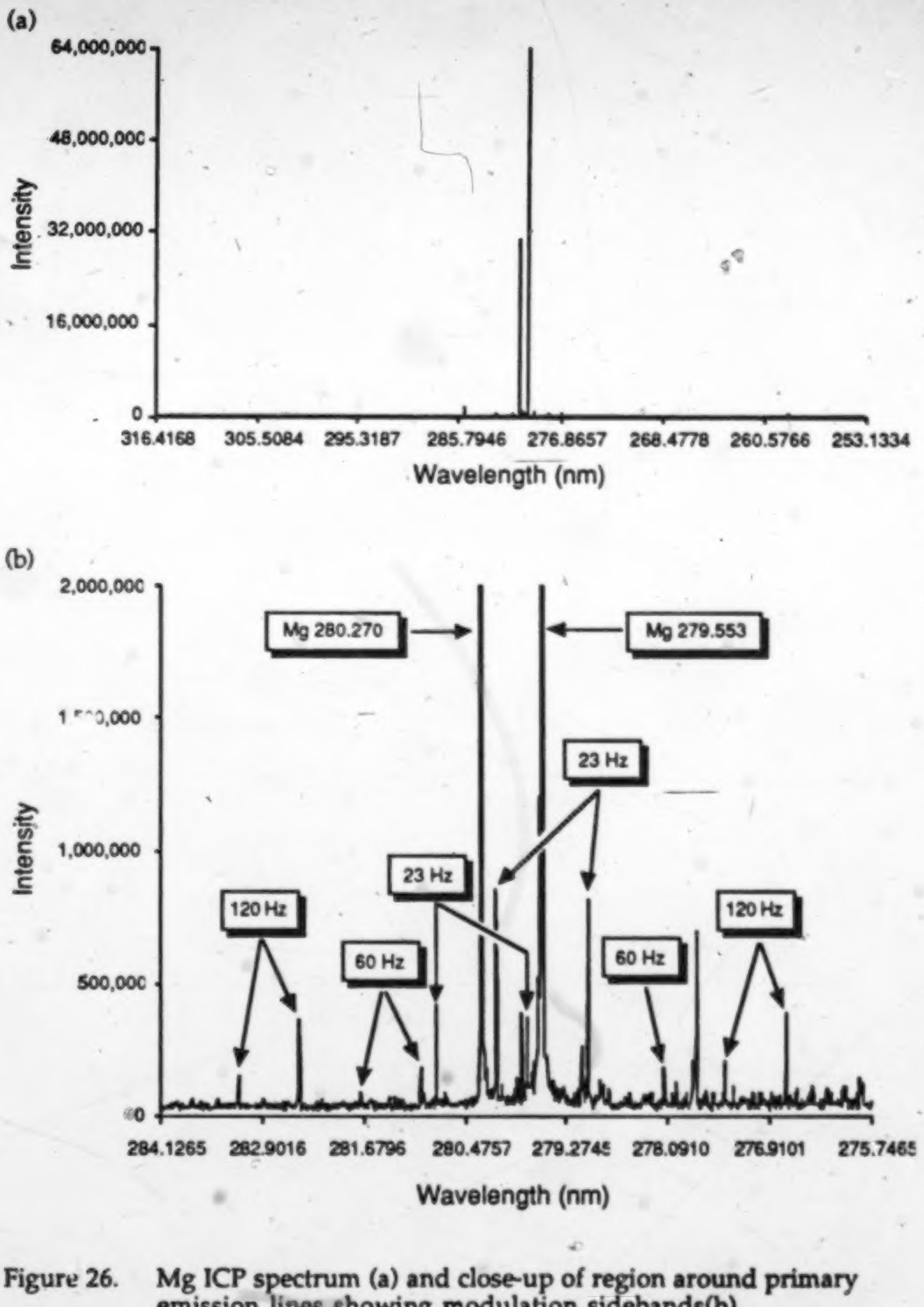


Figure 26. Mg ICP spectrum (a) and close-up of region around primary emission lines showing modulation sidebands(b).

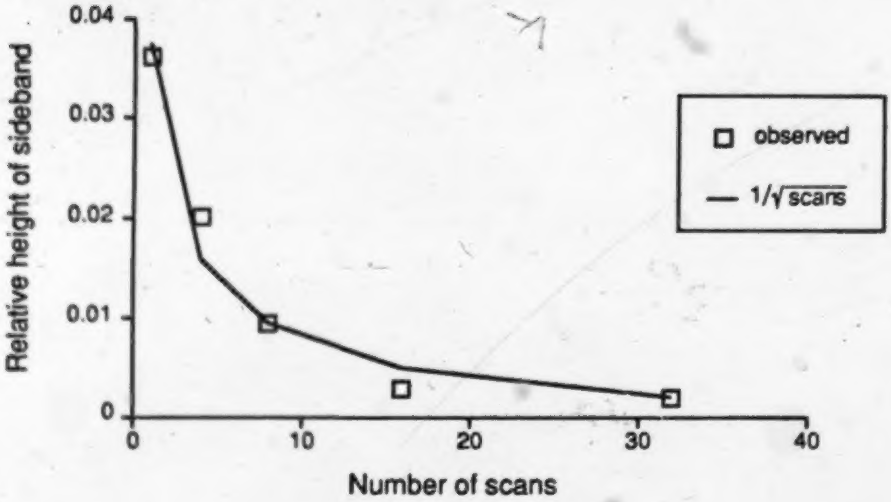


Figure 27. Reduction of modulation sideband intensity by signal averaging showing both observed and predicted values.

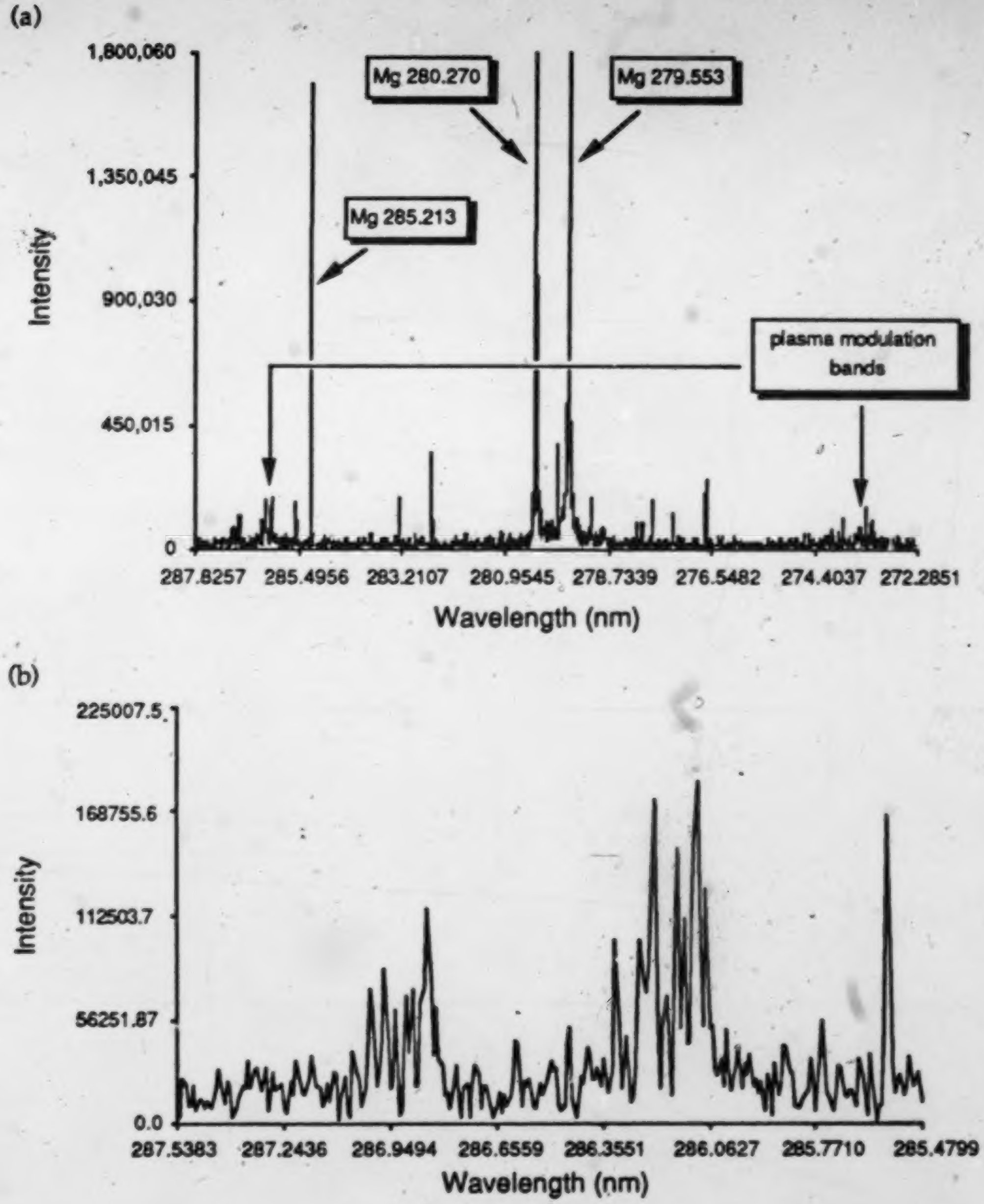


Figure 28. Scale expanded view of a Mg ICP spectrum showing the plasma modulation bands on either side of the parent peaks (a) and a close-up view of the left bands (b).

tion and some electrical changes in the hardware have resulted in the reduction of the 23 Hz and 60 Hz modulation components

4-5. Conclusions

Our interferometer system does indeed provide accurate and precise wavelengths after it has been calibrated using standard wavelengths. For more accurate results the spectra should be interpolated (zero filled) so that the lineshape can be better approximated. Values obtained with our system can be accurate to ±0.00025 nm (±0.06125 cm^{-1} at 200 nm) which is higher than the ±0.001 nm commonly quoted in the literature.

The spurious peaks sometimes found in FTS spectra have been shown to arise from modulation of the interferogram. Either the mirror drive or the emission source can be modulated and the modulation can be either electronic or mechanical, the effects are indistinguishable. The intensities of these modulation sidebands can be reduced through signal averaging. In fact, the presence and relative intensities of modulation sidebands now serve as figures of merit (the fewer and smaller, the better) in constructing the instrument.

Overall it appears that our ICP–FTS system is particularly well suited for qualitative spectrochemical work. It can provide highly accurate wavelength and intensity information spanning a wide spectral region in a short period of time. Wavelength accuracy is important because it allows positive identification of atomic emission lines. Thus, high wavelength accuracy, high spectral resolution, rapid data acquisition rates, and small size are major features of the current system.

Chapter 5

Quantitative atomic spectrochemical performance.

Fourier transform spectrometers have all but totally replaced scanning monochromator instruments in infrared spectroscopy. This is because FTS instruments offer two main advantages over conventional grating systems, they are optically faster and they deliver superior signal-to-noise ratios (SNR's) for the same measurement time. The two factors are known, respectively, as Jacquinot's advantage and Fellgett's advantage and arise from the multiplex nature of the Fourier transform operation.

The improvement in SNR arises from the fact that all wavelengths are measured simultaneously in FTS. This allows an FTS instrument to integrate each spectral element for the entire measurement period, instead of just a small fraction of it as in scanning monochromators. The longer effective integration time increases the SNR because the SNR is proportional to the square root of the integration time.

Unfortunately, this argument assumes that the dominant noise is independent of signal intensity. This holds true for the IR but in the visible and ultraviolet portions of the electromagnetic spectrum the background noise of available detectors can be below the photo noise level and the dominant noise becomes proportional to some power of the signal intensity.

In the case where the dominant noise is photon noise, which is proportional to the square root of the signal intensity, FTS instruments no longer have a SNR advantage over conventional dispersive systems and when the system is source noise limited conventional instruments are better. Winefordner *et al.* [87] have done an extensive theoretical study of SNR performance of FTS instruments in the various noise regimes.

The multiplex disadvantage refers to the cases where conventional instruments offer better SNR performance than multiplex systems such as FTS. The disadvantage results from the distribution of noise from a single localized spectral feature throughout an entire spectrum. This means that the noise from an intense emission line can obscure weak lines in a spectrum and the phenomenon occurs because all wavelengths are measured simultaneously.

In order to determine the suitability of ICP–FTS for quantitative analytical work the noise characteristics, dynamic range, and detection limits of our system need to be studied and critically evaluated.

5-1. Measuring noise

Before one can measure and discuss noise in atomic spectrochemical systems, both noise and the methods for determining it must be defined. A generic definition of noise is, the variation in the measured values of some fixed quantity over a series of repeated measurements. In the case of a single channel spectrometer, this is simply the fluctuation of the signal at the output of the instrument. If the noise is normally distributed, then the standard deviation of many repeated measurements can serve as an estimate of the noise.

For spectra, the situation is more complicated since each measurement is a spectrum composed of many discrete spectral elements. In this case, repeated measurements mean multiple spectra with noise being the variations between the spectra. If each spectral element is considered independently of all others, then the noise distribution in the overall spectrum is obtained.

By acquiring multiple spectra and calculating the standard deviation at each spectral element from the corresponding intensity values in each of the spectra, a standard deviation spectrum is obtained. This spectrum is a measure of standard deviation at each spectral element across the spectrum being measured. An outline of this procedure is shown in Figure 29. In addition, a signal-to-noise ratio spectrum can be calculated by ratioing the mean intensity spectrum with the standard deviation spectrum.

Mean intensity, standard deviation, and signal-to-noise spectra for 10 mg/L of Mg aspirated into an ICP are shown in figure 30. The measurement procedure was the same as that used by Marra and Horlick [21]. We have also standardized on the same definition of a single measurement as them. Each individual spectrum was calculated from 16 co-added interferograms and the mean and standard deviation were calculated from 8 spectra.

If the noise is stationary, i.e. doesn't change with time, it is possible to estimate the amount of baseline noise in a spectrum with only a single measurement. This is done by selecting a region of the spectrum that is devoid of spectral features and calculating the standard deviation of the points within that region.

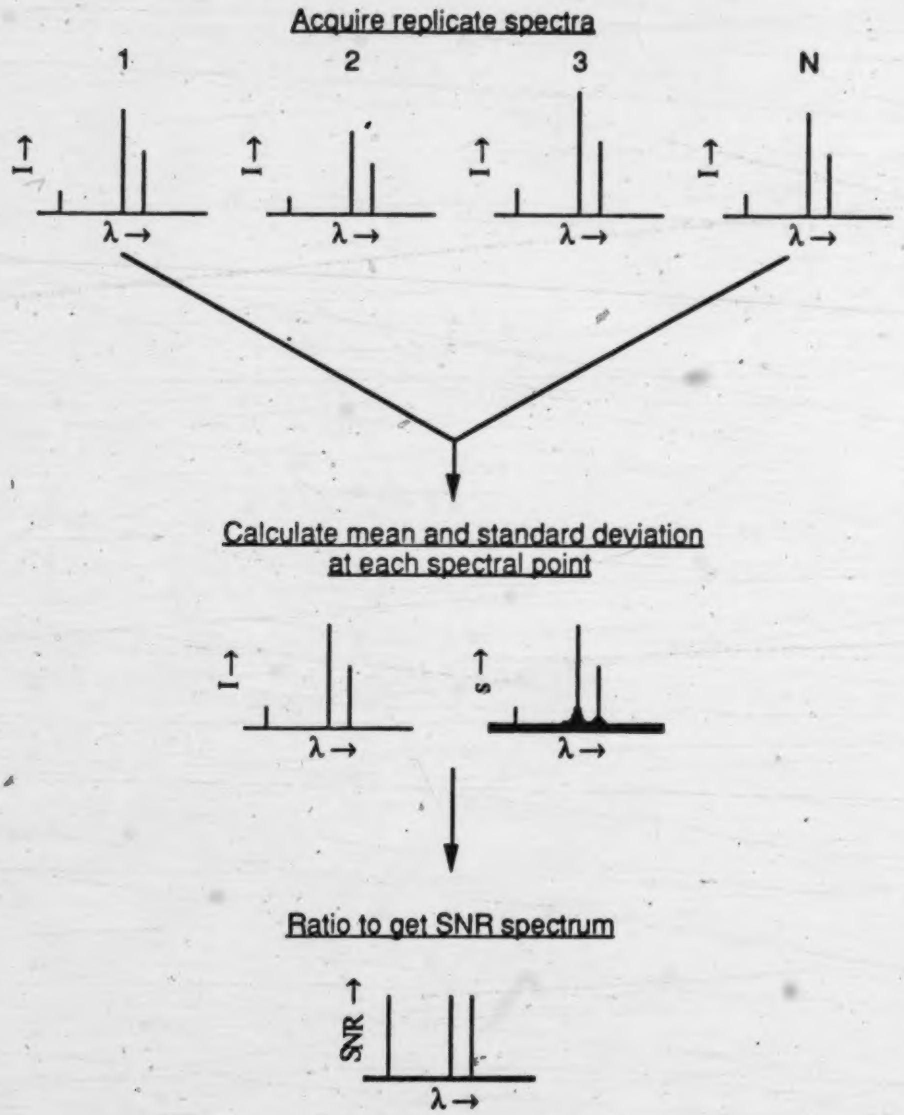

Acquire replicate spectra
1
2
3
N
I →
λ →
Calculate mean and standard deviation at each spectral point
I →
s →
λ →
Ratio to get SNR spectrum
SNR →
λ →

Figure 29. Outline of procedure for calculating standard deviation and signal-to-noise ratio spectra.

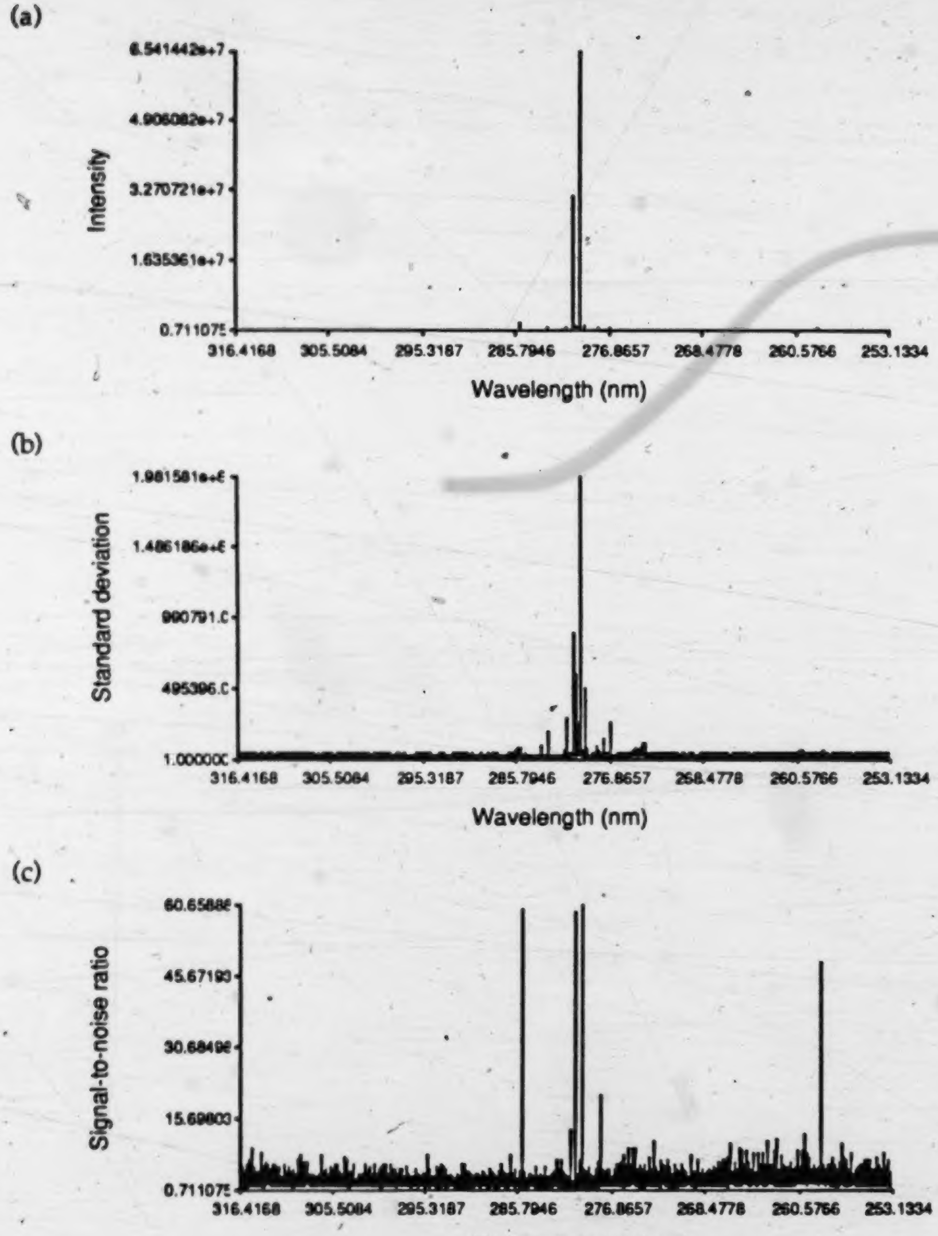


Figure 30. Intensity (a), standard deviation (b), and signal-to-noise ratio (c) spectra of Mg ICP emission. Calculated from 16K aliased interferograms.

For a 100 point region in an 8192 point spectrum of 10 mg/L Mg, both the average value of the standard deviation spectrum of eight runs and the standard deviation of the intensity values of a single run were measured. The region was at approximately 292 nm and was chosen because it contained no spectral features. The average standard deviation was 22,014 and the standard deviation of the intensities from a single run was 21,629 which agrees well with the value obtained from the standard deviation spectrum.

It must be stressed that the noise at the baseline may be different from the noise at a spectral line. In the case of the ICP, noise can be in one of three different regimes (depending upon emission intensity); detector, photon, or flicker noise limited. In the last two cases, the actual amount of noise is proportional to signal level.

Because of the dependence of noise on the signal level in the ICP, the amount of noise at a spectral line cannot be estimated from a single spectrum. In order to determine the SNR at a line, multiple spectra must be acquired and a standard deviation spectrum calculated. Detection limits on the other hand, which are based on the baseline noise, can be estimated from a single ICP spectrum.

The practical advantages to estimating the baseline noise level from a single spectrum is that it reduces the amount of data required for the estimate and it can be calculated faster. The calculation of a standard deviation spectrum requires multiple spectra, which in the case of high resolution spectra can occupy several megabytes of disk storage. The main advantage is convenience, only a single spectrum is required to calculate a detection limit.

5-2. The multiplex disadvantage

Much previous work in assessing the multiplex disadvantage has assumed that the limiting noise is white in nature and only recently has the effect of 1/f noise been considered [88]. Even the very recent work by Hobbs *et al.* [89] assumes white noise in the ICP. This situation exists despite that fact that most real noise sources are non-white because white noise is the easiest to model.

It is possible to expand the modulation sideband model, described in Chapter 4, to include noise sources. For example, white noise can be thought of as the result of an infinite number of modulation frequencies each with a random but statistically similar (i.e. the same mean and variance) amplitude. This conclusion is based on the power spectrum of a white noise source.

For a noisy source Equation 4-16 now becomes:

$$I'(\delta) = A_0 + A_0 \cos(2\pi\delta\nu_0) + \sum_{i=0}^{\infty} A_i \cos\left(\frac{\pi f_i \delta}{\bar{v}}\right) + \frac{A_i}{2} \cos\left(2\pi\delta\nu_0 \pm \frac{\pi f_i \delta}{\bar{v}}\right)$$

Equation 5-1

And the resulting spectrum from a Fourier transform spectrometer will have an infinite number of random intensity sidebands spreading across the entire spectrum. These random sidebands form the noisy baseline of the spectrum. This type of model has also been used by Sing and Hubert [85] and by Voigtman and Winefordner [88].

For the case of 1/f noise, the intensity of the sidebands decrease as they move away from the parent line. This is because the noise power decreases with frequency which is translated by Equation 4-16 into distance from the parent line. The result is that 1/f noise is concentrated around spectral features instead of being spread throughout the entire spectrum like white noise.

In a flicker noise situation the intensities of the sidebands are proportional to the intensity of the parent line, hence they would become larger as the parent line becomes more intense. This is the source of the multiplex disadvantage, where it is possible for the noise from intense lines to completely obscure weak lines in a spectrum.

In the ICP the flicker noise is 1/f in nature rather than white [86], this implies that the noise should be concentrated around the emission lines and that the multiplex disadvantage only occurs near intense lines which are the noisiest. This affect should appear as increased baseline noise near the intense lines that is proportional to the line intensities, regions far from such lines should be unaffected.

To truly test this concept, an experiment similar to the source modulation experiment performed in Chapter 4 could be done using an apparatus described by Voigtman and Winefordner [90] to impart a 1/f fluctuation on a stable source. At this time, this has not been done. All of our noise studies are of ICP emission which should illustrate 1/f noise problems in a qualitative manner.

The concentration of noise near emission lines in ICP–FTS spectra was first observed by Marra and Horlick [21] using standard deviation spectra and we have also observed this. The standard deviation in the vicinity of the Mg 279 and 280 nm emission lines for spectra calculated from 4K interferograms are shown in Figure 31b. The same region of the intensity spectrum is shown in Figure 31a.

The 4K interferograms were obtained by truncating 16K interferograms, this was done to see the effect of resolution on standard deviation spectra and because the original work done by Marra and Horlick used 4K interferograms. The 16K interferograms were sampled using the basic HeNe clock and a solar blind PMT was the detector. Spectra calculated from such interferograms are aliased.

The intensity and standard deviation spectra calculated from the original 16K interferograms are shown in Figures 32a and 32b. The important thing to note is that noise (standard deviation) is greatest in the immediate vicinity of the emission lines and that higher resolution further concentrates the noise.

A new feature appears in the standard deviation spectrum that is calculated from 16K interferograms when close attention is given to the region in the immediate vicinity of a spectral line (see Figure 33b). The actual standard deviation at each emission line is a local minimum and the noise is greatest on the shoulders of the line forming what appears to be a doublet.

Another interesting observation is that this only occurs at real spectral lines, noise peaks are not split. This phenomenon is seen in Figure 33a, where peaks corresponding to 23 Hz modulation sidebands are seen between the two doublets corresponding to the 279 and 280 nm emission lines. The 23 Hz modulation is the result of a small oscillation in the velocity control circuitry of the interferometer which produces spurious peaks on either side of real spectral lines.

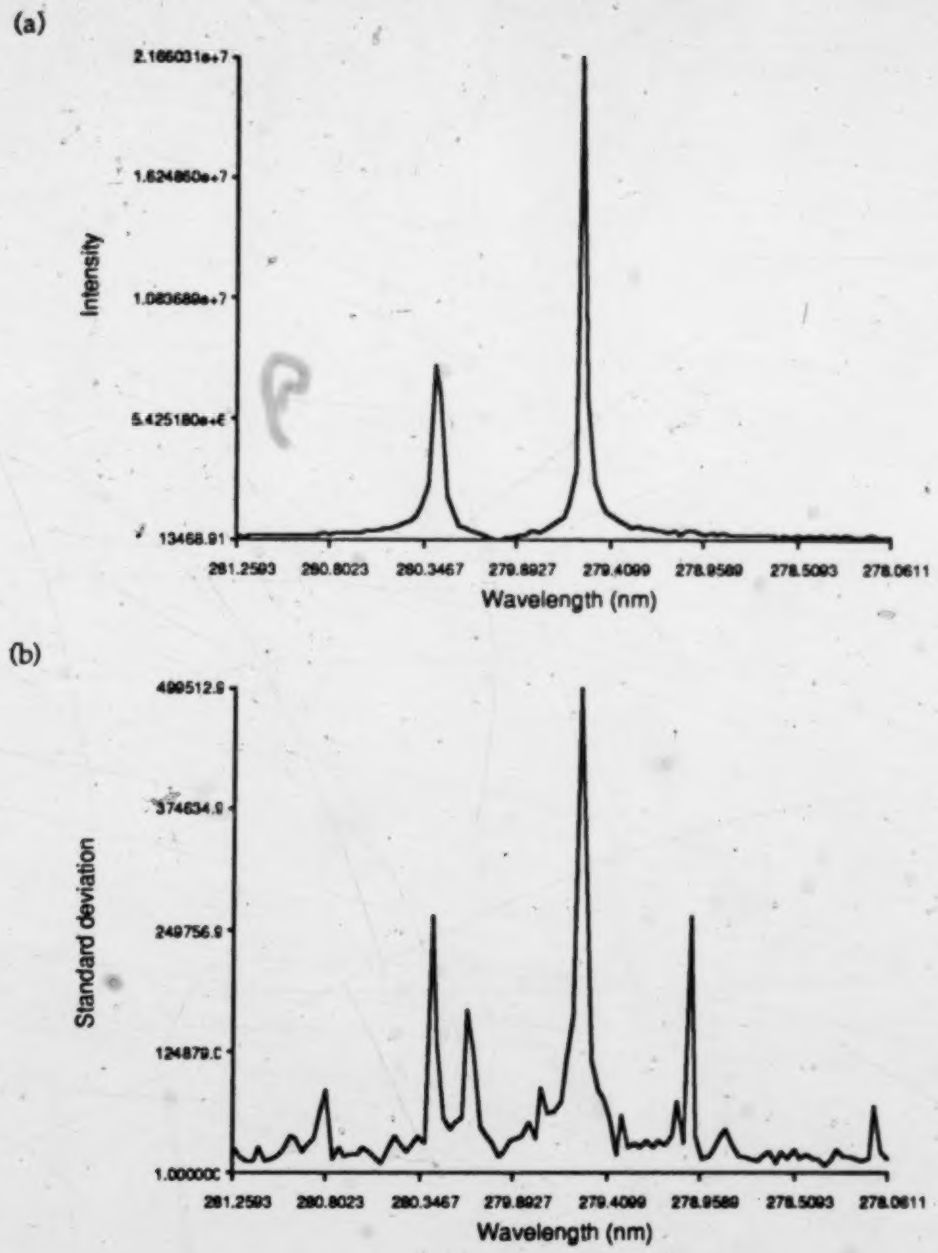


Figure 31. Close-up of Mg spectra showing intensity (a) and standard deviation (b) spectra. From 4K 10 mg/L Mg ICP interograms sampled with laser clock (aliased).

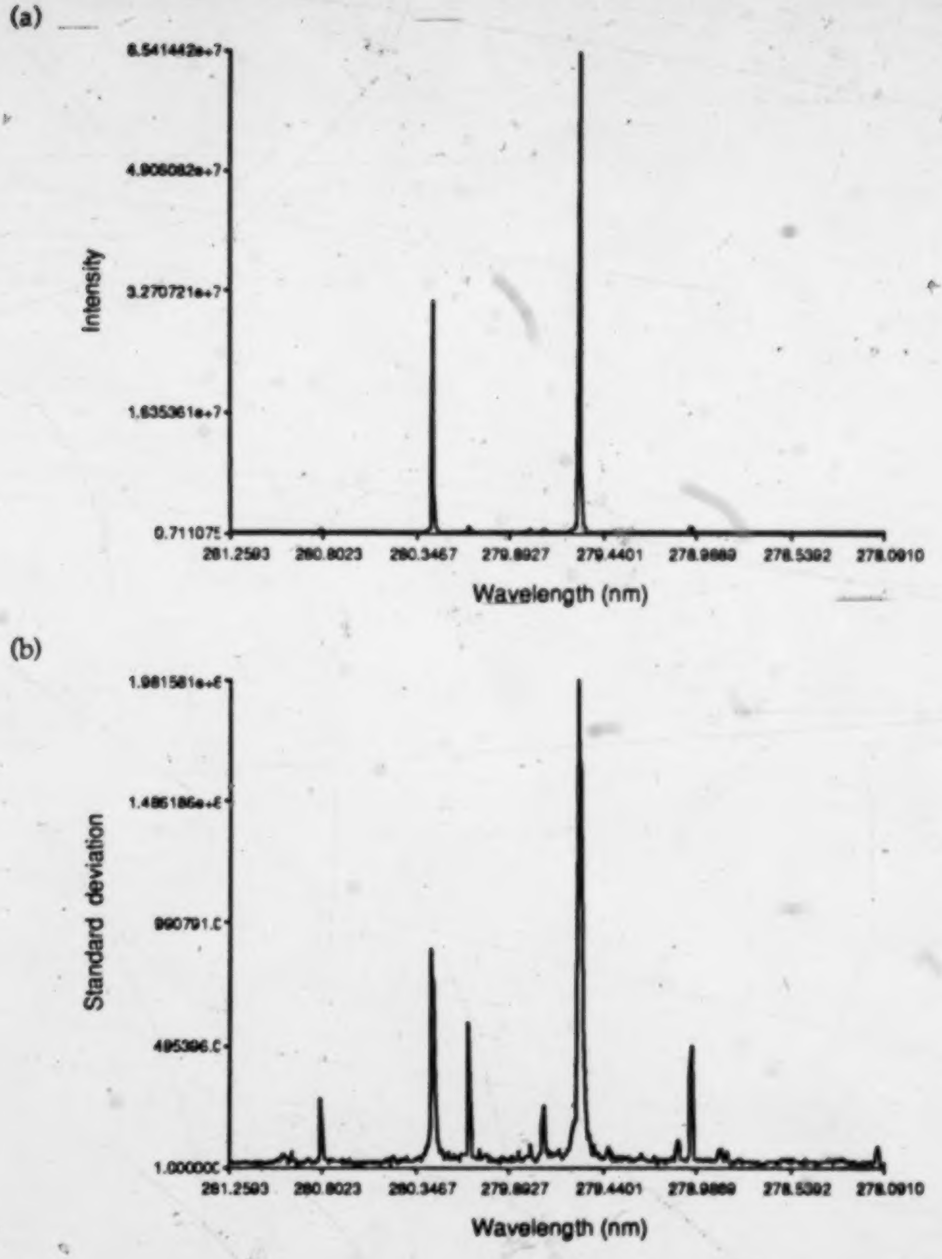


Figure 32. Close-up of Mg spectra showing intensity (a) and standard deviation (b) spectra. From 16K 10 mg/L Mg ICP interograms sampled with laser clock (aliased).

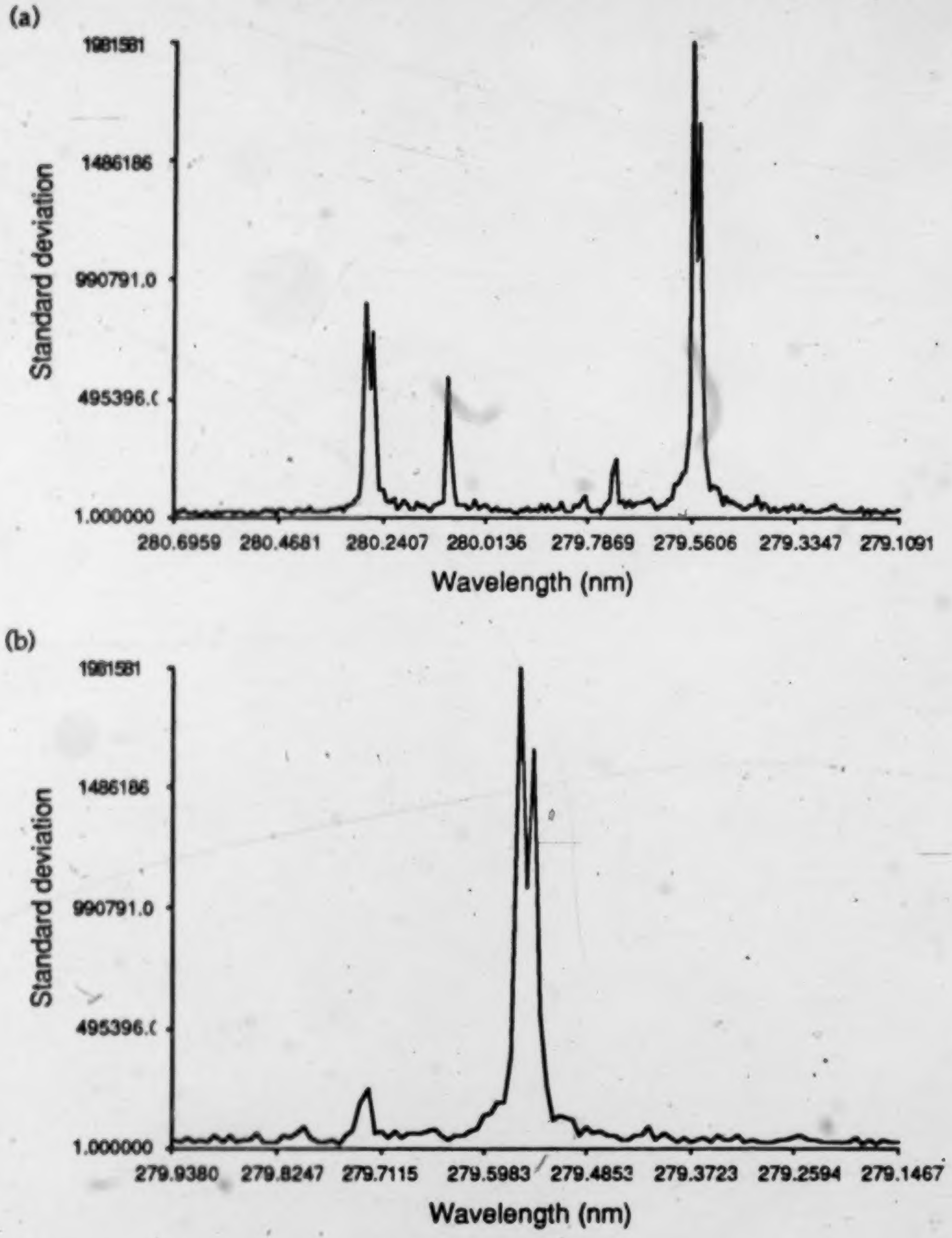


Figure 33. Closer views of Mg standard deviation spectra calculated from 16K interferograms.

The effect of analyte concentration on the baseline noise was studied next by recording the spectra of several different concentrations of Mg. The baseline noise in the ICP spectra was estimated by calculating the standard deviation of baseline intensities in a 100 point region at ~284 nm in each spectrum. This was done instead of calculating standard deviation spectra because it could be done with only a single spectrum. The spectra were calculated from 32 co-added 16K interferograms sampled with the basic HeNe clock.

The linear dependence of the baseline noise on analyte concentration for the resulting aliased spectra is shown in Figure 34. This implies that the multiplex disadvantage is distributing the noise from peaks to the baseline and we appear to be in a flicker noise limited situation. Again, the results are similar to those obtained by Marra and Horlick [21].

Our current instrument can sample an interferogram at various sampling frequencies so it is possible to study the effects of reduced aliasing. A single spectrum of 100 mg/L Mg was obtained from 32 co-added 64K interferograms sampled with the 4× clock. The spectrum has the same spectral resolution as those obtained in the previous experiment but has four times the free spectral range (now ~160 nm). Because a solar blind detector is used, this results in fully de-aliased spectra.

A scale expanded view of the de-aliased spectrum is shown in Figure 35. The fluctuations in the baseline values appear to decrease with decreasing wavelength. This decrease was not observed in aliased spectra because the noise doesn't decrease appreciably over the free spectral range of the aliased spectrum. This was confirmed by comparing the baseline of a region in the

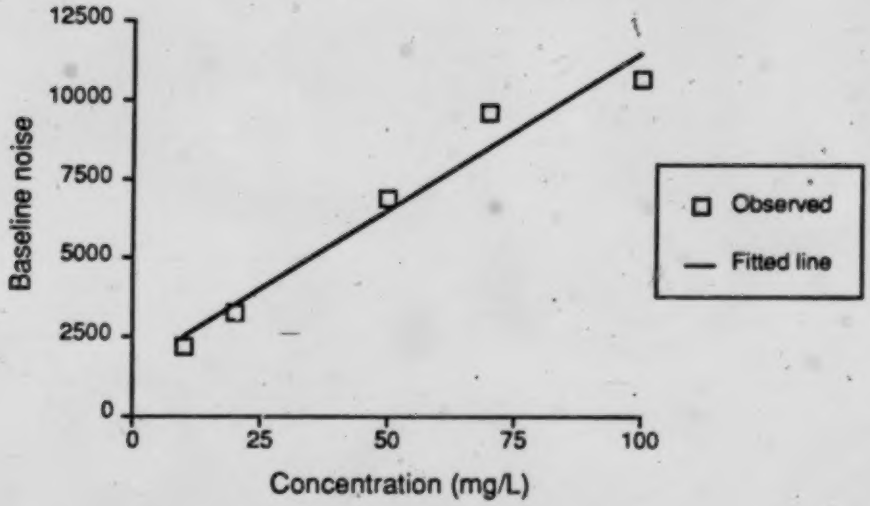


Figure 34. Plot of background noise as a function of Mg concentration in an aliased spectrum.

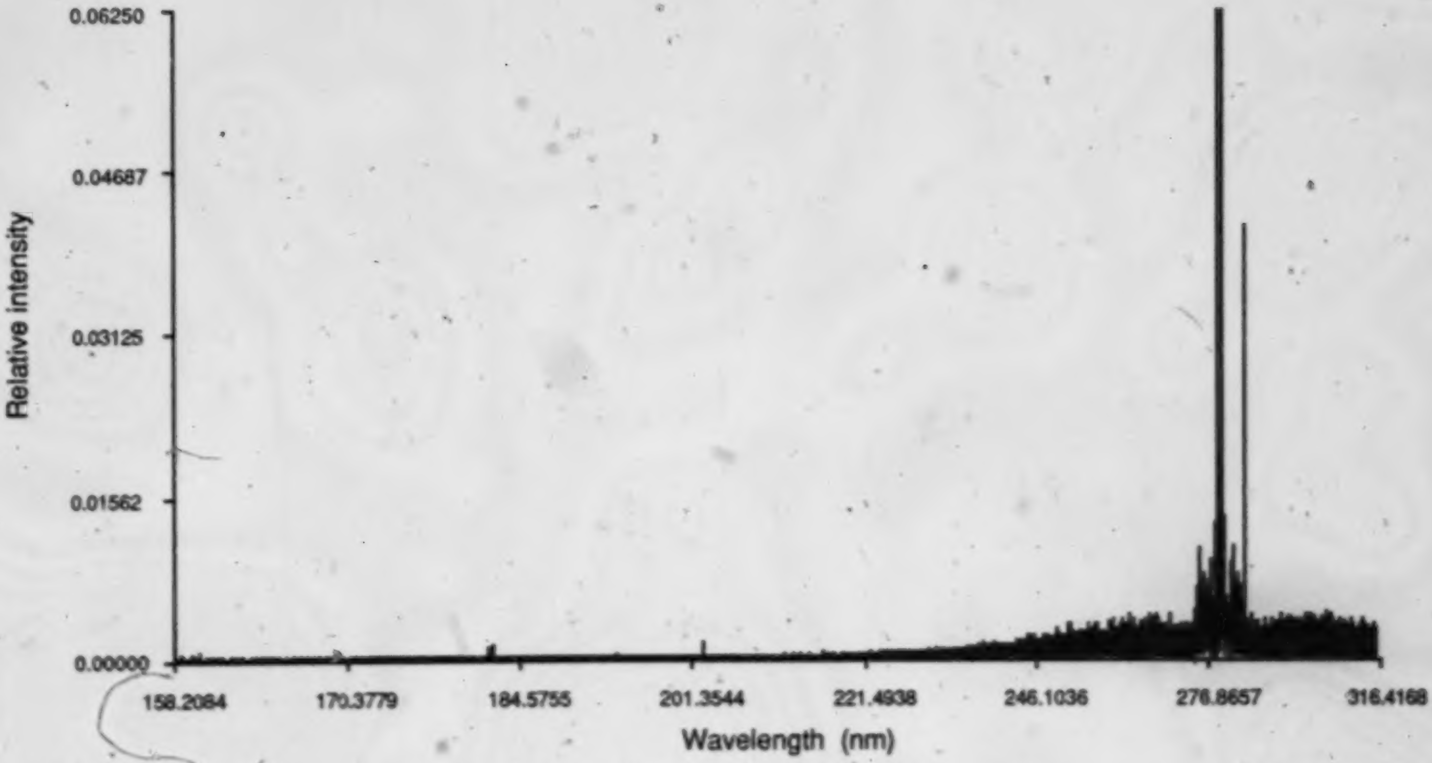


Figure 35. Scale expanded, de-aliased ICP spectrum of Mg showing distribution of baseline noise.

de-aliased spectrum corresponding to the free spectral range of an aliased spectrum (Figure 36a) with an actual aliased spectrum (Figure 36b). The baseline noise in this region of the de-aliased spectrum is now similar to the aliased spectrum in that it now appears to be constant.

The plasma modulation components in the de-aliased spectrum also appear closer to the spectral lines. This is due to a change in the instrument hardware that occurred between the dates when the data were acquired. The speed of the interferometer moving mirror was doubled at that time, this halved the separation between a spectral line and its associated modulation components. Note that the wavelength axes of the two spectra increase in opposite directions because of aliasing.

This leads us to question whether the multiplex disadvantage does distribute noise over an entire spectrum. The effect of analyte concentration on baseline noise both near and far from spectral lines in de-aliased spectra is shown in Figure 37 for Mg. The noise near the spectral lines is calculated at 265 nm while the far noise is calculated at 190 nm. As expected, the noise near the peaks shows a definite relationship with concentration but the noise far away is essentially constant.

In order to observe the effects of the multiplex disadvantage, a simulated sample consisting of 1 mg/L of Zn in 1000 mg/L of Fe was prepared. Zinc was chosen because its major analytical line (213 nm) is well separated from all iron lines, even in an aliased spectrum. This can be illustrated by comparing a zinc spectrum (Figure 5-38a) with iron (Figure 5-38b). Complete spectra are shown, the wavelength axes differ because they correspond to different aliasing regions. The 16K interferograms were acquired

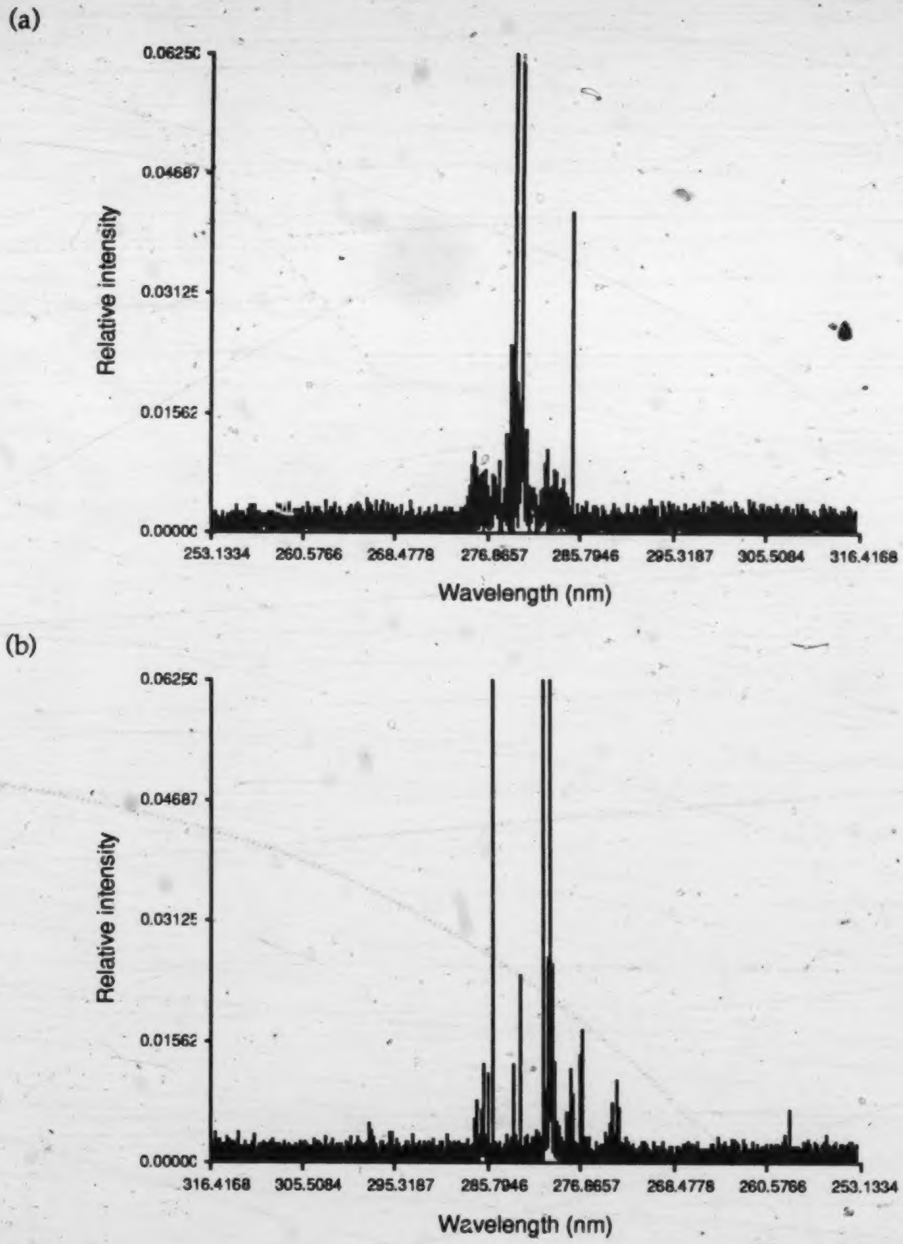


Figure 36. Region of de-aliased spectrum (a) corresponding to an aliased spectrum (b).

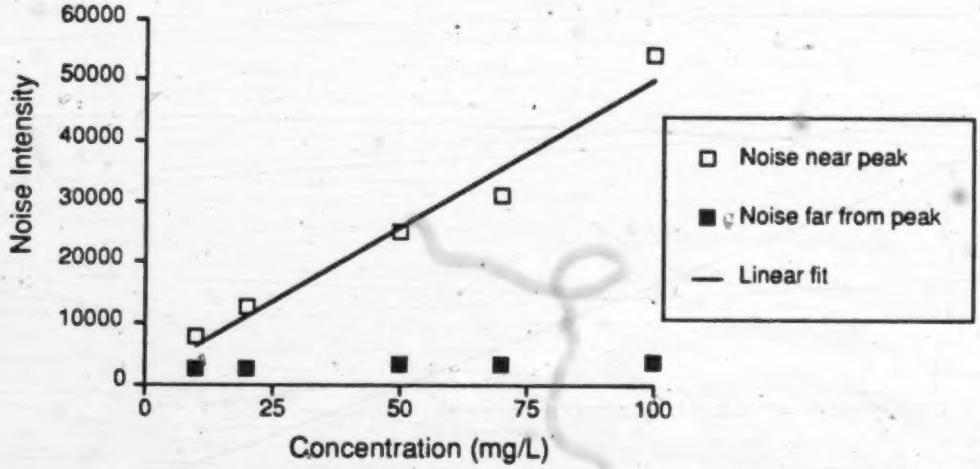


Figure 37. Plot of baseline noise as a function of Mg concentration from a de-aliased spectrum.

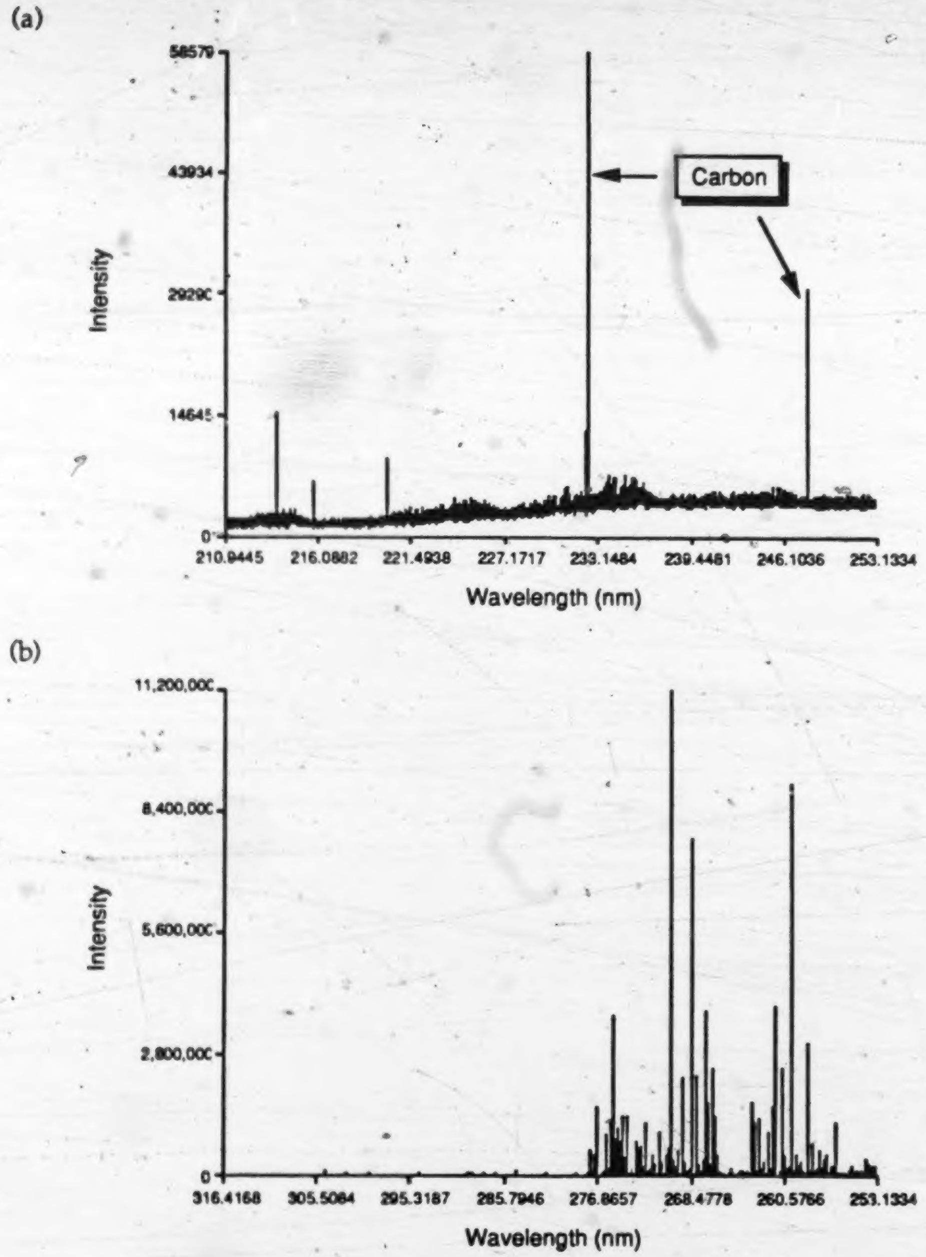


Figure 38. Aliased ICP spectra of Zn (a) and Fe (b) showing the relative distribution of their lines.

with the basic HeNe clock under identical conditions. Carbon lines result from CO_2 impurities in the Ar gas.

The spectral region around the Zn 213 nm line for 1 mg/L Zn, 1000 mg/L Fe, and 1 mg/L Zn in 1000 mg/L Fe are shown in Figure 5-39. The Zn signal is well resolved from its background noise (Figure 5-39a) but the background noise for Fe (Figure 5-39b) is much higher than for Zn because of the multiplex disadvantage. When the two species are combined the noise from Fe almost totally swamps out the Zn signal despite the large spectral separation (Figure 5-39c). This demonstrates the dire consequences of the multiplex disadvantage.

The problem persists in de-aliased spectra because the Zn 213 nm line is still near enough to strong Fe lines in the region around 238 nm to be affected by their noise.

5-3. Detection limits

The detection limit (dl) of a species is formally defined as the smallest amount of the species that produces a signal that can be distinguished from a blank. The major difficulty with this definition is determining how a signal can be distinguished from a blank. Usually, a statistical test is applied to determine this criteria.

The IUPAC method for calculating detection limits defines the limit of detection c_L as:

$$c_L = \frac{k\, s_B}{m} \qquad \text{Equation 5-2}$$

(a)

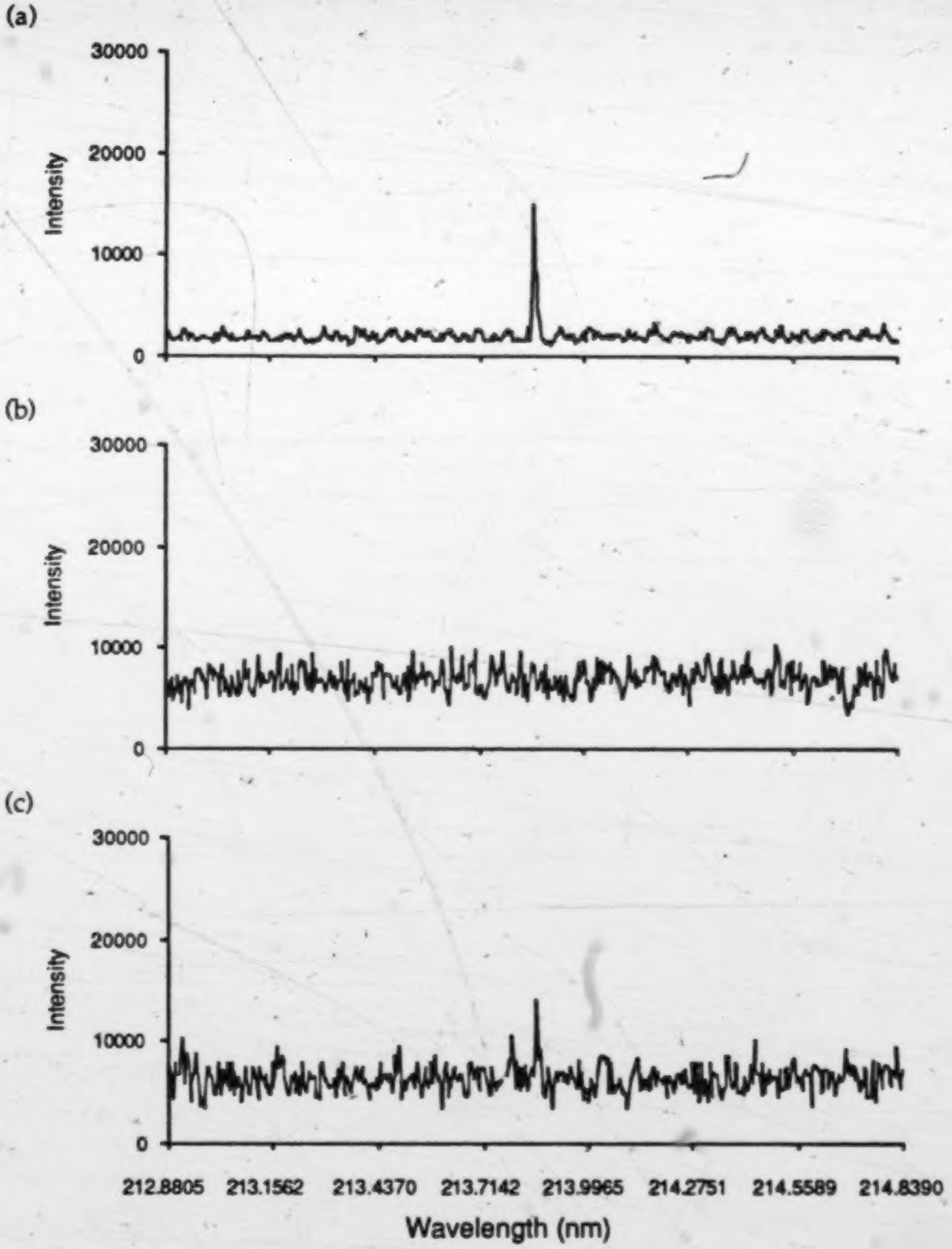


Figure 39. The effect of 1000 mg/L Fe on the aliased spectrum of 1 mg/L Zn. Spectra of 1 mg/L Zn (a), 1000 mg/L Fe (b), and 1 mg/L Zn + 1000 mg/L Fe (c).

Where s_B is the standard deviation of the blank, m is the slope of the calibration curve, and k is an integer. k is usually chosen to be 3 since this assures a <0.14% probability of a false positive if the background noise is normally distributed [91]. This relation also assumes that all measurements are blank subtracted.

Because of its importance in calculating detection limits, it is imperative that the standard deviation of the blank be measured properly in order to assure accurate limits of detection. There are several methods of obtaining the information necessary to calculate a detection limit depending on the analytical technique and type of instrumentation.

1) Single channel detector.
   s_B is calculated from multiple readings of blank. Signal strength is measured for a known concentration to calculate dl.
   Alternatively, a calibration curve could be constructed and used instead.

2) Complete spectrum available.

   A) Peak height method.
      Peak height at analytical line is either used directly or to construct a calibration curve. s_B is calculated using one of the following methods:

      i) Calculate s_B from multiple blank spectra at the analytical line.

      ii) Calculate s_B from a region of baseline in the blank spectrum that is close to the analytical line.

iii) Calculate s_B from a region of baseline in the analyte spectrum.

B) Peak area method.

Peak area of analytical line is either used directly or to construct a calibration curve. s_B is calculated using one of the following methods:

i) Calculate the area of the baseline at the analytical line in a blank spectrum using the width of the analytical line. Calculate standard deviation from multiple blank spectra.

ii) Divide a region of baseline in the blank spectrum that is close to the analytical line into units with the same width as the analytical line. Measure the area of each unit and calculate the standard deviation between them.

iii) Divide a region of baseline in the analyte spectrum that is close to the analytical line into units with the same width as the analytical line. Measure the area of each unit and calculate the standard deviation between them.

Methods 2.A.i and 2.B.i are the equivalent to method 1 but are not often used because acquiring multiple blank spectra is time consuming. When s_B is measured from a region in a single spectrum, baseline measurements are made explicitly in regions with no spectral features, uniform baseline, and the same noise level as at the analytical line. Methods 2.A.iii and 2.B.iii assume that the noise in the baseline doesn't change with concentration and that the calibration curve is linear.

Several types of instruments are capable of measuring complete spectra. They are: rapid scanning monochromators, photodiode array spectrometers, CCD array spectrometers and Fourier transform spectrometers. All the methods of determining detection limit can be used with these instruments. Though direct reading polychromators are restricted to method 1 it is possible for them to integrate peak areas by setting the entrance slit so that the entire peak width passes through the spectrometer and onto the detector.

For pure Mg, using the 279.553 nm line, a detection limit of 10.3 ng/mL was obtained on our ICP–FTS system while for Fe 238.207 the detection limit was 142 ng/mL. Such performance is one to two orders of magnitude worse than that of dispersive instruments and is consistent with previous work by Stubley and Horlick [18].

These detection limits were calculated from calibration curves constructed from 32 scan, 16K aliased interferograms. The blank standard deviation was calculated from a nearby region of baseline in a blank solution and peak height values were used in the calibration curve (method 2.A.ii).

Peak areas were not used because of the narrowness of our peaks (7 points full width) and the large amount of noise on the shoulders of the peaks. These factors would lead to noise accumulating faster than the signal when integrating over the peak area. The use of peak areas would therefore result in higher detection limits than peak height.

Because the baseline standard deviation may vary in a spectrum, the baseline region used to calculate s_B should be as close to the analytical line as possible. Also because of the multiplex disadvantage, detection limits determined for ICP–FTS from pure aqueous solutions may not be applicable to real samples. The matrix can have a severe degrading effect on the real detection limit.

The data acquisition parameters, number of scans and interferogram length, were chosen as typical values that maximize sample throughput. It is possible to achieve better detection limits through signal averaging longer and increasing the interferogram length as these factors increase the measurement time.

Since the measurement time is proportional to the length of the interferogram, it is possible to equate interferogram length with measurement time. Because noise accumulates as the square root of the measurement time while signal is directly proportional to measurement time, the detection limit will decrease as the square root of the increase in measurement time.

Long and Winefordner [91] also discuss using a more rigorous approach to calculating detection limits that also takes into account errors in the slope and intercept of the analytical curve. Using this approach our values for Mg and Fe become 11.4 ng/mL and 146 ng/mL, respectively. These values are not much different from those obtained without taking into account errors in the analytical curve which implies that our analytical curve is highly linear with a low uncertainty over the two to three orders of magnitude that they cover.

5-4. Dynamic range

Another consideration for applying FTS to ICP–OES is its dynamic range. This is an important figure of merit since optical emission from the ICP is usually linear over six orders of magnitude change in concentration. An ideal measurement system for the ICP should be able to match this kind of performance, currently no systems do.

Dynamic range is defined as the ratio of the maximum measurable signal intensity to the magnitude of the baseline noise. On our FTS system it was measured by setting the amplifier gains so that an interferogram from the solution under study fills the greatest possible portion of the ADC range without overflowing. Scans were accumulated by adding successive interferograms. After a spectrum was calculated from the acquired interferogram(s) the baseline noise (standard deviation of the baseline) was ratioed into the most intense spectral line.

Our current data system uses a 12-bit ADC which limits the maximum single spectrum dynamic range to 4096. However, by accumulating scans it should be possible to improve on this by a factor of $\sqrt{N}$ where N is the number of scans. The effect of accumulating scans on dynamic range, baseline noise, and signal intensity are all shown in Figure 40. The measurements were performed on 1000 mg/L Mg using 16K interferograms sampled with the basic HeNe clock. Interferograms composed of 16, 32, 64, and 128 co-added scans were acquired.

The effect of the number of scans on the dynamic range is shown in Figure 40a. The expected behavior is that dynamic range should increase as the square root of the number of scans. This is shown by the line in the graph (log–log slope of 0.5). However, the measured dynamic range is lower than expected at both the high and low ends of the range of scans.

The reason why dynamic range is less than expected at the low end of Figure 40a is that the baseline noise is higher than expected. The effect of the number of scans on the baseline noise is shown in Figure 40b. The expected behavior is that baseline noise should increase as the square root of the number of scans. This is shown by the line in the graph (log-log slope of 0.5). All the data points but the lowest fall on the predicted noise line. It would appear that below 30 scans the baseline noise does not decrease but rolls-off to some constant value. This indicates that quantizing noise in the ADC is the primary source of noise in this region.

The loss of dynamic range at the high end of Figure 40a can be attributed to the loss of signal intensity due to instrument drift. The effect of the number of scans on the signal intensity is shown in Figure 40c. The expected behavior is that signal intensity should increase linearly with the number of scans. This is shown by the line in the graph (log-log slope of 1). As the number of scans increases the deviation of the measured intensity from the expected line also increases. The deviation is always negative. This observation is consistent with drift in the alignment of the interferometer which would cause a decrease in signal intensity.

As a consequence, there is currently a limit to the amount of dynamic range that can be achieved on our instrument by accumulating interferograms. It appears to be not a fundamental limit but an engineering problem. The interferometer is currently not in a temperature controlled environ-

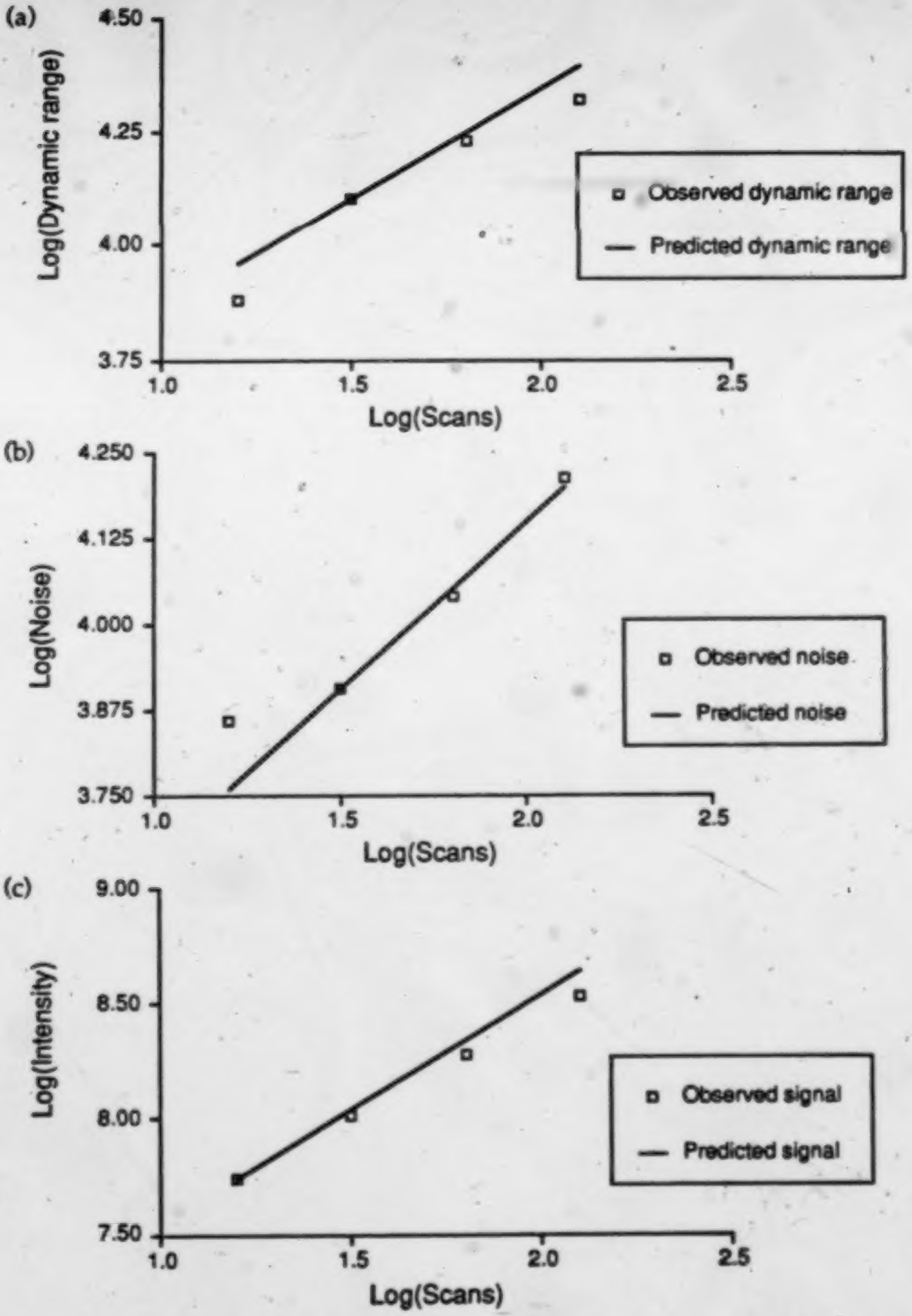


Figure 40. The effect of accumulating interferograms on dynamic range (a), noise (b), and signal intensity (c).

ment. Changes in room temperature cause subsequent changes in the alignment of the interferometer optics.

The dynamic range of a spectrum is also affected by the length of the interferogram from which it is calculated. This is because the Fourier transform is an integral transform, the input data is multiplied by a weighting factor and is summed over the data range to produce an output value. Extending the length of the interferogram in effect increases the integration range. This is consistent with the interpretation of Hobbs *et al* [89]. Given an interferogram that does not decay, the resulting spectral values will be proportional to the length of the interferogram.

In actual practice, the natural linewidth and the instrumental line function cause the interferogram to decay in intensity. This puts a limit not only to the resolution of the resulting spectrum but also to its dynamic range, Figure 41a shows this effect. A 64K interferogram consisting of the sum of 32 scans was obtained for a 1000 mg/L solution of Mg. The 32K, 16K, 8K, and 4K interferograms were obtained by truncating. The dynamic range for the 8K data is abnormally low because the spectrum lacked a data point at the 279.553 nm position, this lead to an erroneously low intensity reading.

The observed effect of interferogram length on dynamic range is due solely to the dependence of signal intensity on the interferogram length, the amount of baseline noise doesn't vary (see Figure 41c). The intensity of the signal increases with interferogram length but eventually reaches a plateau (see Figure 41b) which leads to a similar effect in dynamic range (Figure 41a). The reason for this is obvious when we look at the interferogram (Figure 42).

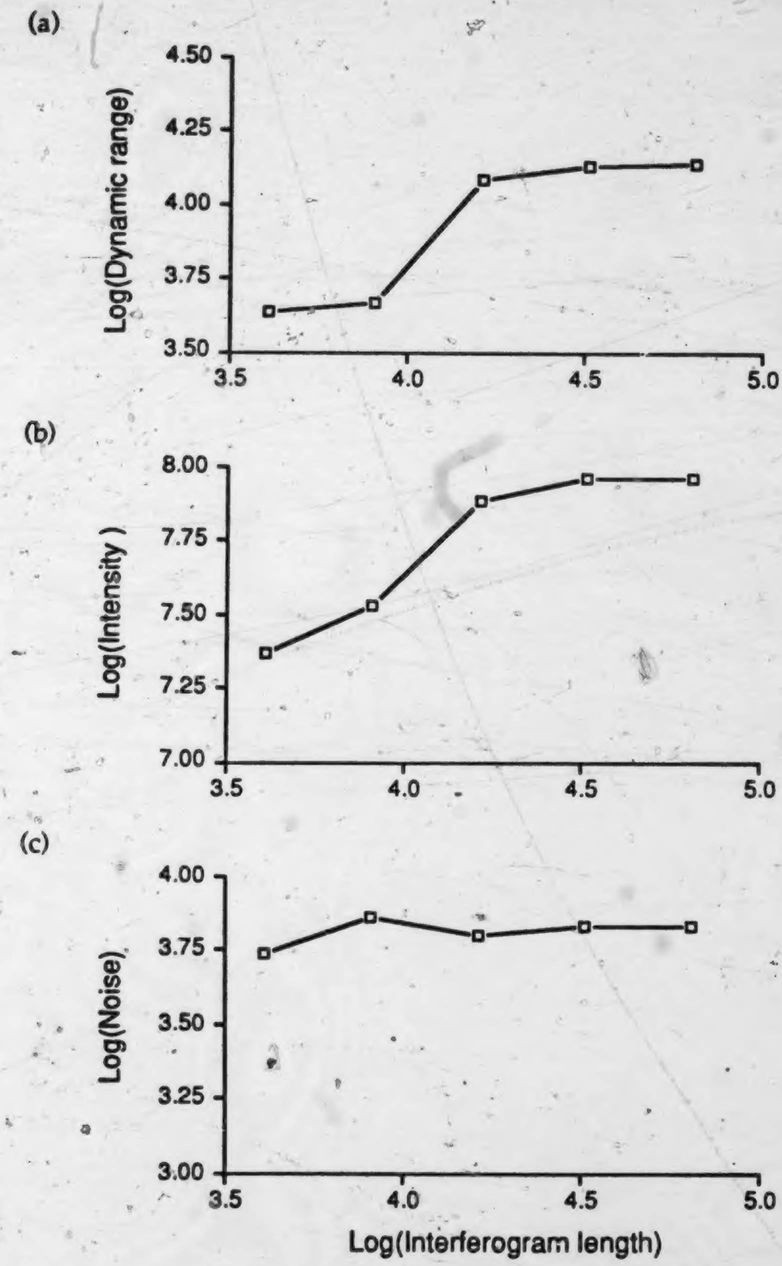


Figure 41. The effect of interferogram length on dynamic range (a), signal intensity (b), and baseline noise (c).

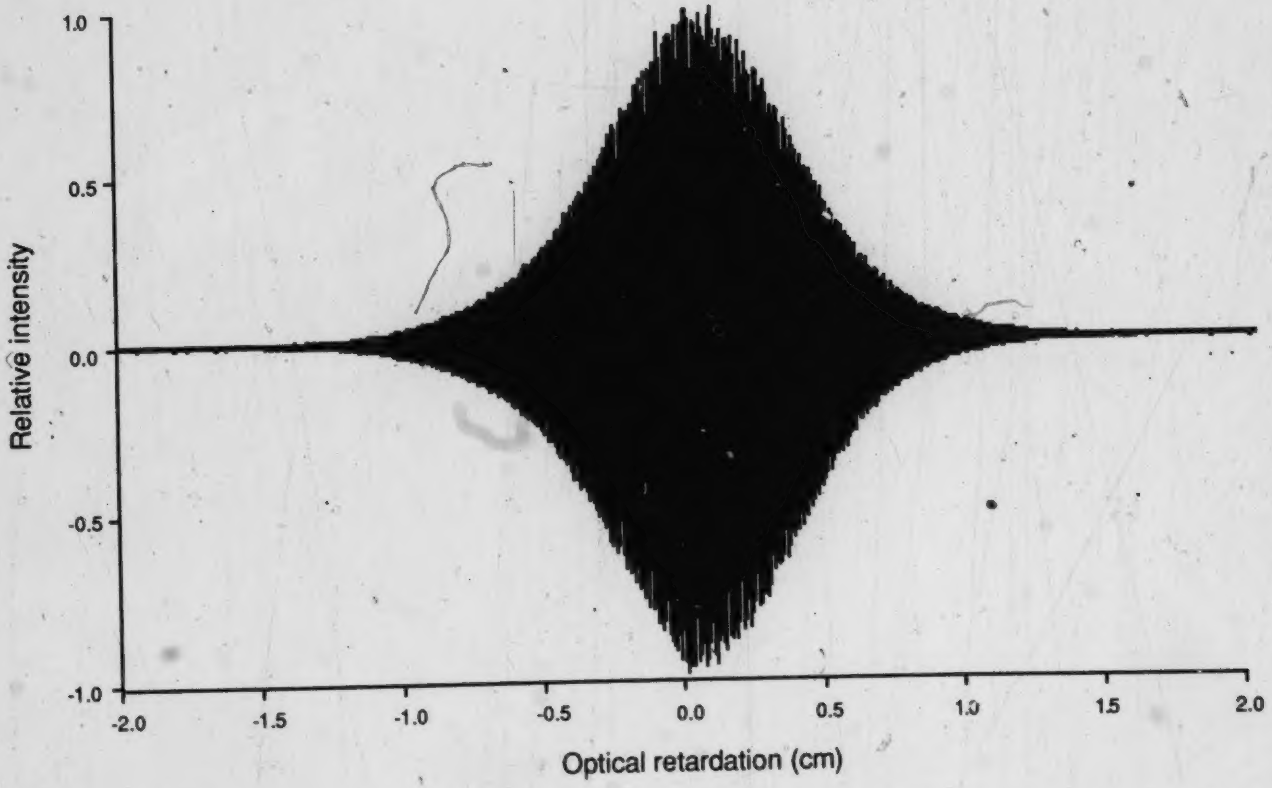


Figure 42. 64K interferogram of Mg ICP emission.

The interferogram intensity decays to almost zero by ±1.5 cm optical retardation and between these points and when the moving mirror turns around for a 64K interferogram (±2 cm optical retardation) only noise is being measured. For shorter interferograms the signal still decays at the ends but the drop is usually small enough that a gain in signal intensity can still be realized.

It is this decay in interferogram intensity that sets a working limit to the length of the mirror drive in an interferometer. Use of a longer drive, i.e. higher resolution, will not result in any improvement in performance since there is no longer any information to be measured.

We can achieve typical dynamic ranges of around four orders of magnitude, depending on the number of scans accumulated and the length of the scans, with a 12 bit ADC. The dynamic range observed at 128 scans is less than expected due to lower than predicted intensities and is due to drift in the alignment of the interferometer. The dynamic range performance of our ICP–FTS system for simple spectra is comparable to conventional dispersive based PDA instruments [92], however the multiplex disadvantage will degrade the dynamic range for complex spectra.

5-5. Conclusions

The multiplex disadvantage is a severe problem for ICP–FTS, the noise from large amounts of matrix elements will readily obscure the emission from the trace elements under investigation. It is highly doubtful that it is possible to overcome the multiplex disadvantage without sacrificing some of the benefits of FTS. Therefore, improvements in the noise characteristics of the ICP are required when FTS is used as a measurement system.

Based upon detection limits that are at least an order of magnitude higher than those attained with regular dispersive instruments it is apparent that ICP–FTS is not a trace technique. A more realistic application of ICP–FTS would be as a survey technique to characterize samples before attempting trace analysis. This type of application would take advantage of the high throughput, simultaneous wavelength coverage, and accurate wavelengths afforded by FTS.

Despite its poor trace performance, ICP–FTS is capable of performing more than adequately at higher concentrations. Its dynamic range is comparable with that of dispersive instruments and its calibration curve is highly linear with little uncertainty. Some noise and drift problems still exist but it should be possible to overcome them through better engineering.

One obvious means of improving performance is to use higher quality analog electronics in the signal path and a higher resolution ADC. Because both weak and strong lines are measured simultaneously, FTS performance can only benefit from increasing the number of quantizing bits available. FTS cries out for the use of high speed 16 bit ADC's. As an aside, in order to match the dynamic range of the ICP a 20 bit ADC is required.

Chapter 6

A spectral atlas of ICP atomic emission lines for selected elements.

Fourier transform spectroscopy is especially attractive for compiling spectral atlases because of its high resolution, wavelength accuracy, and speed of acquiring entire spectra. A definite need that has been expressed in the analytical atomic emission community is for a high quality compilation of atomic emission lines in a form that is easily accessed [93]. Existing atlases either consist of tables of prominent lines [94] or photoreproductions of strip chart recordings [95]. The problem is that tables lack many lines and it is difficult to extract precise information from strip chart recordings.

An atlas that is specific to the ICP is required because a number of lines are seen that are not observed in arc and spark sources. These lines have never before been compiled and remain unassigned. We have attempted to comprehensively assign all lines that are visible in the acquired spectra, the remaining lines have been compiled into tables of mystery lines requiring further study. Because of time limitations, we have elected to concentrate primarily on elements and spectral regions where information is lacking [93].

Another major advantage of FTS is that the entire spectrum is available in a machine readable form. This facilitates distribution and is essential for developing databases for automated spectral interpretation systems. When used in conjunction with highly interactive graphical software it is possible to treat the spectra like electronic stripchart recordings.

6-1. Experimental parameters

All interferograms were acquired from a Plasma Therm 2500 ICP using our custom built Michelson interferometer. The ICP RF forward power was set at 1.5 kW and the Ar gas flows were 13.5 lpm (outer), 1.0 lpm (intermediate), and 0.8 lpm (central/nebulizer). The nebulizer flow translates into a sample uptake of 1 ml/min into a concentric glass nebulizer. This is a natural uptake rate, no pump is used to deliver the sample solution to the nebulizer. A 30 cm focal length quartz lens is used to collimate light from a spot 15 mm above the load coil through a 8 mm aperture in the interferometer.

The detector was a Hamamatsu R166 solar blind PMT operated at 660V. The PMT output was amplified with a Keithley model 427 current amplifier and bandpass filtered though a Krohn-Hite model 3343 analog filter. The Krohn-Hite filter consists of two independent 48 dB/octave high pass/low pass configurable filters and is also capable of supplying up to +40 dB of extra gain. The Krohn-Hite was set up to pass the region from 17 to 37 kHz. This roughly corresponds to the spectral region from 170 to 370 nm on our instrument and is outside the 180 to 330 nm response of the R166.

Both high resolution (0.5 cm^{-1}) aliased and medium resolution (2 cm^{-1}) de-aliased spectra were acquired for each element. They were calculated from 64K interferograms sampled with the basic HeNe and 4× clocks respectively. The de-aliased spectra were used to confirm the lines found in the aliased spectra which were then used for wavelength assignment.

The acquired interferograms were directly transformed without any additional processing. In the case of the high resolution interferograms, the natural decay of the interferogram was sufficient to not require any additional apodization.

The sample solutions consisted of either 100 or 1000 mg/L of each species in single component aqueous solutions. The solutions were prepared from 1000 mg/L commercial stock solutions. In general, the concentrations used were an order of magnitude greater than that required by a conventional dispersive instrument [95]. The elements used and their exact concentrations are listed in Table 8. In addition to elements where existing data is incomplete, some elements commonly present in steels were also measured.

Some lines for C and Si have also been tabulated, C is sometimes present as CO_2 contamination in the Ar plasma gas and Si results from attack on the glass nebulizer/spray chamber assembly when the sample is prepared with HF. Mg lines are also listed, a 10 mg/l solution of Mg is used to set-up the instrument.

6-2. Choosing the number of scans

When choosing the number of scans to signal average the trade-off is between dynamic range and the probability of instrument drift. To maximize dynamic range a large number of scans must be averaged, however this requires a lengthy measurement period which increases the probability of instrumental drift. It is instrumental drift that sets a practical limit to the amount of dynamic range that can be gained by signal averaging on our system. A point is eventually reached where the signal intensity has decayed sufficiently due to drift that further signal averaging results in no gain in dynamic range.

Element	Concentration (mg/L)
Cd	100.0
Co	100.0
Cr	99.9
Cu	100.6
Fe	100.0
Hg	997
La	1004
Mn	100.0
Mo	100.5
Ni	100.0
Pb	1006
Sn	993
Ta	1000
Th	992
U	990
V	100.1
W	1000
Zn	100.0
Zr	100.6

Table 8. List of elements and the exact concentrations used in acquiring their spectra.

As a compromise, the spectra for our spectral atlas were calculated from 64 co-added interferograms. Co-addition differs from signal averaging in that the resulting summed interferogram is not divided by the total number of interferograms that were summed. This number of scans, in addition to increasing the dynamic range of the resulting spectrum, also significantly reduces the intensity of any modulation sidebands, thereby reducing the probability of seeing spurious peaks.

6-3. Line assignment

Initial line lists with tentative wavelengths were automatically generated by computer from the calculated spectra. These lists were then manually reviewed for accuracy.

We have attempted to identify all observed lines using lines tabulated in Zaidel [96]. The criterion for assigning an observed line to a literature line is based on wavelength. If the observed and literature wavelengths are within a wavelength window corresponding to ±2 spectral points (±0.002 nm at 200 nm for a spectrum calculated from a 64K aliased interferogram) of each other then they are considered the same. This takes into account both the uncertainty in the observed value and any errors in the literature value.

False positives, spurious lines, were minimized by extensive signal averaging and cross checking the aliased and de-aliased spectra. We have also only tabulated those lines which have intensities greater than 10 baseline standard deviations. In some cases, such as U where the lines have very low intensities, lines between 5 and 10 standard deviations were also tabulated.

Careful attention has also been given towards the possibility of 23 Hz modulation bands at the base of intense lines.

6-4. Unassigned lines

Because of its unique nature, some ICP lines are not observed in arc and spark spectra and are therefore missing from the early spectral line tables made with these sources. A graphic example of this situation was presented by Blades [97]. This outlines one of the major deficiencies of existing line tables when applied to ICP work.

After all the known lines have been assigned there remain some unassigned spectral lines. The unassigned lines for each element were compared with prominent lines of other elements to check for possible contamination. Through this process we discovered Si lines in our W and Hg spectra. This was not surprising since the W solution contained 1% HF, and a standard glass nebulizer and spray chamber was used. The Hg solution was run immediately after the W, hence some cross contamination occurred.

6-5. Results

A typical spectrum, a de-aliased Ta spectrum, is shown in Figure 43a. Because of the scale, the spectrum is highly compressed and individual lines are not discernible. An ~2 nm portion of the Ta spectrum is shown at a more useful scale in Figure 43b.

The lines for each of the elements have been compiled into a database using Reflex Plus for the Macintosh from Borland International and are summarized in Appendix E. Reflex Plus was chosen because it uses the rela-

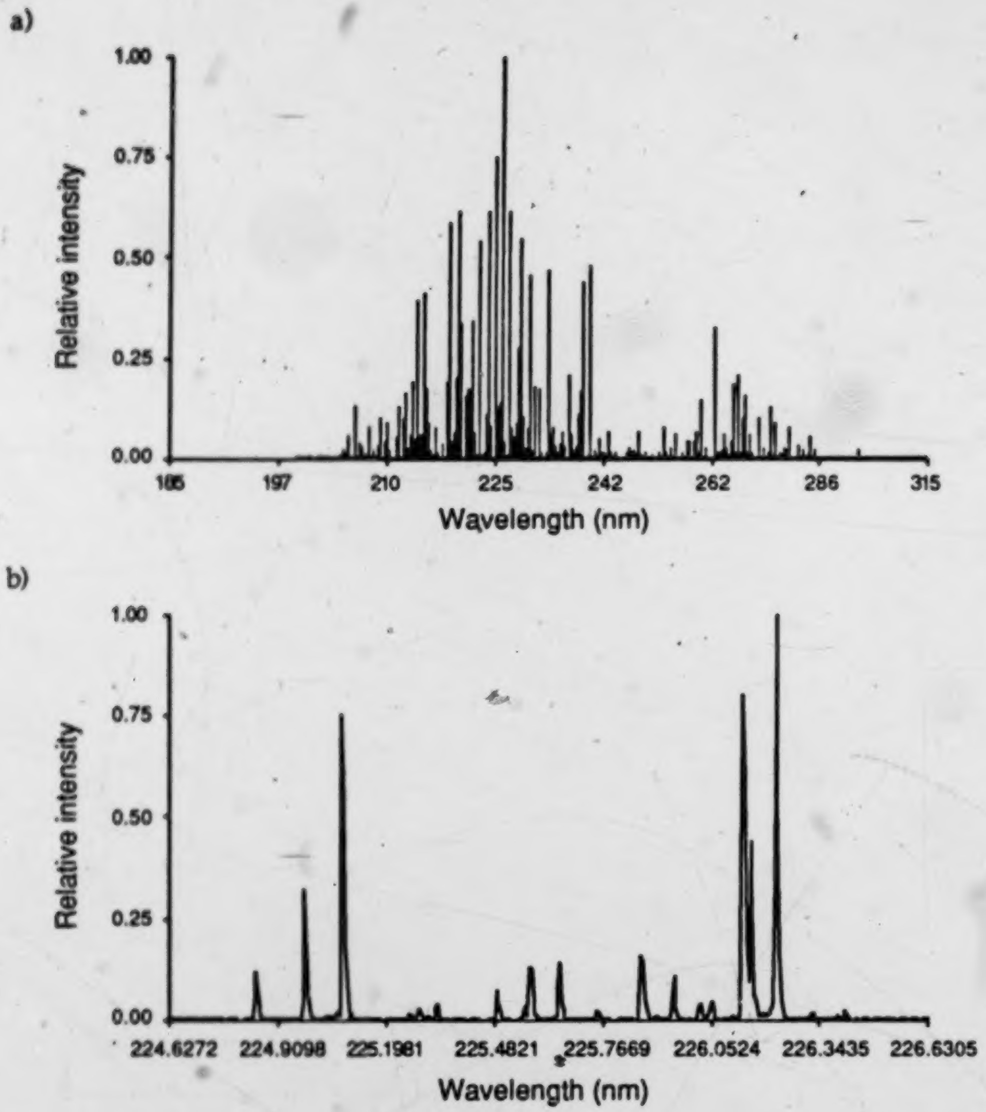


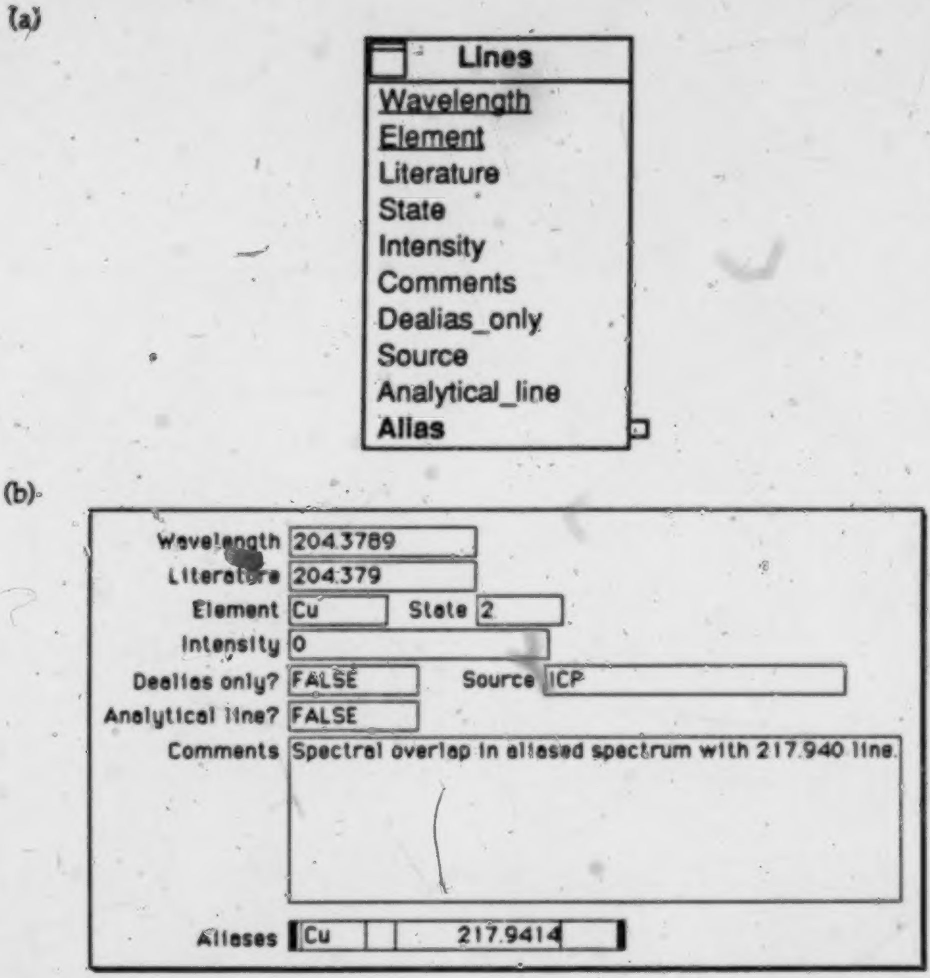
Figure 43. The ultraviolet ICP spectrum of Ta (a) and an ~2 nm close-up of the region containing the most intense emission lines (b).

tional model, is flexible, and the database structure can be altered at any time. The major drawback of Reflex Plus is the lack of a programming language.

Currently each line record in the database contains fields for observed wavelength, literature wavelength, element, state, relative intensity, the source(s), flags denoting whether it is an analytical line and whether the line is only seen in a dealiased spectrum, and a comment field. In addition, an alias field is used to link lines that overlap when aliased (this is why a relational model is necessary). The database is sorted using both observed wavelength and element, this helps resolve possible overlaps. The overall view is shown if Figure 44a and a sample record is shown is Figure 44b, a summary of the entire database is given in Table 9.

A literature wavelength of zero means that the line has not been assigned and an intensity of zero means that it is a minor line in which the relative intensity has not been calculated yet. Wherever possible, the wavelength and intensity values were measured from a high resolution spectrum in order to obtain the most accurate values. For those lines which overlap in the aliased spectrum, the de-aliased values were used. In some cases, lines were only seen in the de-aliased spectrum, these are denoted by a TRUE value for the "Dealias only?" flag, and their intensity values are less reliable.

An example of one of the summary tables in Appendix E is reproduced here. The table for Sn (Table 10) has been chosen as a typical example. The summary tables contain the measured wavelength, relative intensity, element, state, and literature wavelength. For lines below 200 nm, the measured wavelengths are air values while the literature wavelengths are vacuum.



Lines

- Wavelength
- Element
- Literature
- State
- Intensity
- Comments
- Dealias_only
- Source
- Analytical_line
- **Alias**

Wavelength 204.3789
Literature 204.379
Element Cu State 2
Intensity 0
Dealias only? FALSE Source ICP
Analytical line? FALSE
Comments Spectral overlap in aliased spectrum with 217.940 line.
Aliases Cu 217.9414

Figure 44. The overall view of the spectral line database (a) and a sample line data record from Reflex Plus (b).

Element	Total lines	Unassigned lines	Overlapped lines
C	2	0	0
Cd	3	0	0
Co	80	2	2
Cr	81	0	0
Cu	35	0	2
Fe	75	0	2
Hg	8	6	0
La	37	19	0
Mg	6	0	0
Mn	37	8	0
Mo	83	0	0
Ni	92	3	0
Pb	19	0	0
Si	13	0	0
Sn	24	2	0
Ta	161	113	0
Th	79	38	0
U	86	29	0
V	80	0	2
W	89	1	0
Zn	3	0	0
Zr	55	0	4
Totals 22	1148	221	12

Table 9. Overall summary of atomic lines database.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
189.9242	0.1794	Sn	II	189.989
209.1583	0.0602	Sn	I	209.158
211.3929	0.1204	Sn	I	211.393
215.0830	0.0624	Sn		215.084
215.1499	0.2069	Sn		NLV
219.4486	0.1223	Sn	I	219.449
219.9339	0.2043	Sn	I	219.934
220.9651	0.3842	Sn	I	220.966
224.6041	0.6920	Sn	I	224.605
226.6008	0.0645	Sn	I	226.604
226.8921	0.6084	Sn	I	226.891
228.6672	0.0984	Sn	I	228.668
231.7233	0.2082	Sn	I	231.723
233.4805	0.2123	Sn	I	233.480
235.4836	0.8662	Sn	I	235.484
236.8216	0.0636	Sn		NLV
242.1690	0.2475	Sn	I	242.170
242.9478	0.4228	Sn	I	242.949
254.6535	0.0902	Sn	I	254.655
257.1578	0.0923	Sn	I	257.158
266.1228	0.0939	Sn	I	266.124
270.6486	0.8663	Sn	I	270.651
283.9961	1	Sn	I	283.999
286.3294	0.3140	Sn	I	286.332

Table 10. Sn ICP emission lines.

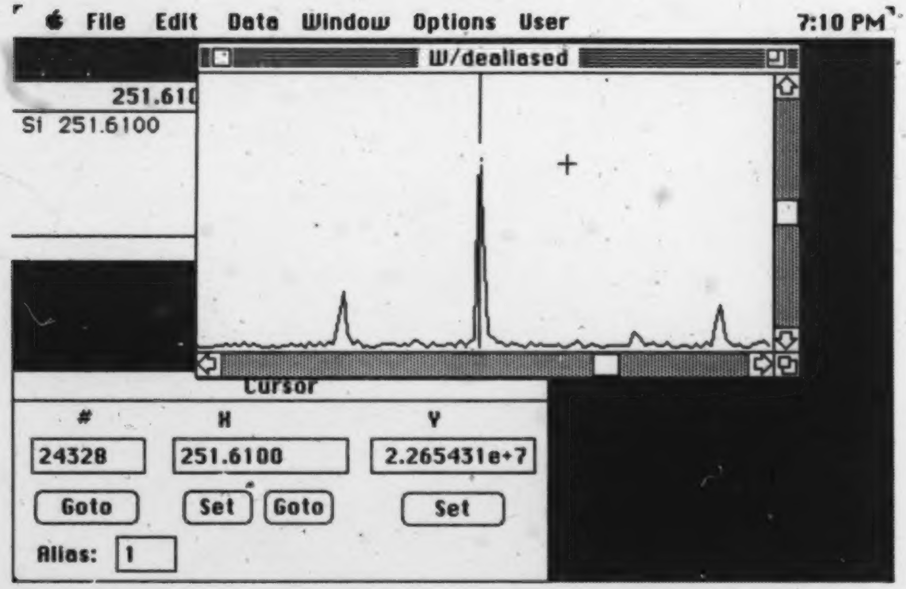
The intensities are relative to the most intense line observed for each element. Where the state entry is empty the line has either not been assigned to a transition or has been assigned to more than one transition.

The data has been summarized in tables because it is difficult to present the complete spectra, or even significant portions, in a useful format in hard-copy. A complete medium resolution (2 cm^{-1}) de-aliased spectrum, plotted out, would be over 42 feet long. For the sake of giving the reader a feel for what the spectra look like, a condensed de-aliased spectrum of each of the elements in Table 8 is included after the element tables in Appendix F. The raw data is available in machine readable form on Macintosh format 3½" 800K floppy disks along with the program to read and display them.

6-6. Conclusions

One practical use of an online atomic spectral database is as an aid in spectral interpretation. A researcher would be able to set a cursor over an unfamiliar peak and ask the computer to lookup possible lines in its database. This has been implemented and an example is shown in Figure 45 where a Si line has been found in a W spectrum. The program searches its database for all lines within a pre-specified window around the cursor position. The advantage over manually looking up the line in one of the standard references is speed, this is especially important if there are many lines to be identified.

Another application would be to subtract stored single element spectra from a multi-component spectrum. This "spectral stripping" could be used to simplify sample spectra to make interpretation easier. Spectral stripping is


File
Edit
Data
Window
Options
User
7:10 PM
W/dealiased
Cursor
Goto
Set
Goto
Set

251.61
Si 251.6100

24328
X 251.6100
Y 2.265431e+7

Alias: 1

Figure 45. A sample screen showing the use of an online database to identify a Si line in a W spectrum.

only practical using FTS because of its absolute wavelength accuracy. The disadvantage is that the stripping process increases the amount of background noise in a spectrum. An alternative would be to use noise-free synthetic reference spectra generated by a spectral simulation program [98].

Currently we are still unable to cleanly "strip" spectra, artifacts remain at the shoulders of subtracted peaks. This is mostly due to difficulties in reproducibly aligning the interferometer fixed mirror. Our current interferometer system must be aligned prior to use and continually re-aligned during operation, this alignment is done manually by the operator. The major problem is thermal drift, the interferometer not being in a temperature controlled environment. If the reference and sample spectra are acquired during the same session over a short time interval, it is possible to do spectral stripping.

The only disadvantage of using FTS to compile an ICP spectral atlas is the multiplex disadvantage. The increased baseline noise obscures very weak lines, especially those near intense lines. For instance, the unassigned Mn line mentioned by Blades [97] is not observed with our instrument.

In addition, more lines are observed in a de-aliased spectrum than in the corresponding aliased spectrum. This is because de-aliased spectra usually have a lower level of baseline noise than aliased spectra. When a spectrum is aliased each of the overlapping sections are added on top of each other, this also causes their variances to be summed giving rise to a net increase in observed noise.

Additional lines tend to be found below 250 nm in de-aliased spectra, a typical example is Pb which is summarized in Table 11. Lines with wave-

Line (nm)	Observed in: aliased spectrum	de-aliased spectrum
205.327	–	yes
216.999	yes	yes
220.351	yes	yes
223.743	–	yes
239.379	yes	yes
240.195	–	yes
241.174	–	yes
244.384	–	yes
244.619	–	yes
244.638	–	yes
257.726	yes	yes
261.365	yes	yes
261.418	yes	yes
266.317	yes	yes
280.200	yes	yes
282.319	yes	yes
283.307	yes	yes
287.332	yes	yes

Table 11. Lines observed in aliased and de-aliased ICP spectra of Pb.

lengths longer than 250 nm are generally observed in both spectra. The reason for this is that the far ultraviolet region is quieter than the longer wavelength regions (see Figure 46), so weaker lines can be observed in this region if other regions are not aliased on top of it.

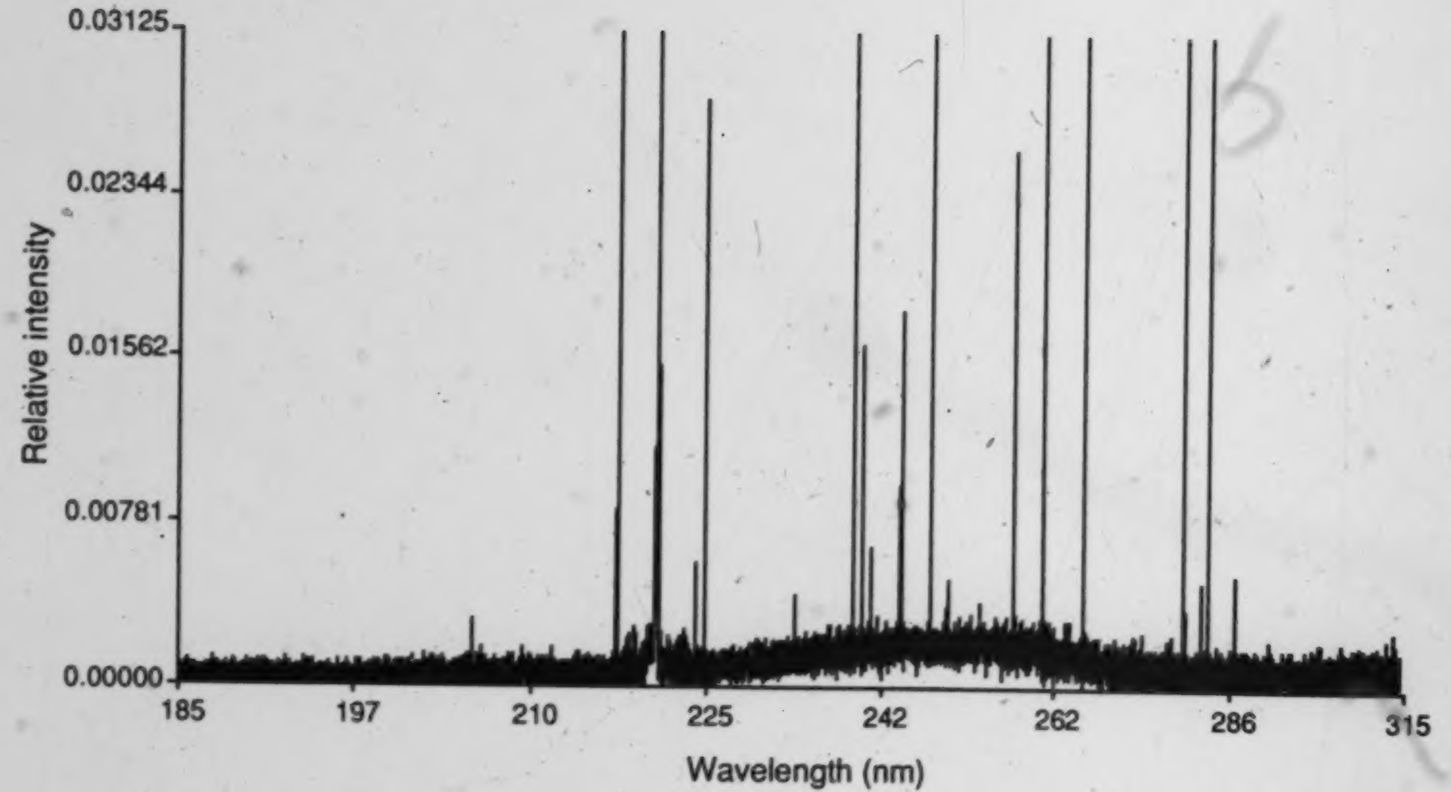


Figure 46. Scale expanded, de-aliased ICP spectrum of Pb showing distribution of baseline noise.

Chapter 7

Automatic identification of spectral lines in ICP–FTS spectra.

Because of the large amount of data that can be acquired using ICP–FTS, it is desirable to automate its interpretation. One way to accomplish this is to have the computer extract all the spectral lines in a spectrum and present them to the user along with tentative assignments. This alleviates the tedious task of examining the entire spectrum manually and ensures that no spectral features are over-looked.

Atomic spectra are particularly amenable to automated analysis because of the exact wavelengths of atomic lines. This allows lines to be assigned simply by matching wavelengths with a database of known atomic lines. The assignments are tentative at this point because aliasing and spectral interferences can lead to multiple assignments for a single line. It is also possible that spurious spectral features can also be assigned giving rise to false elements.

It is at this point where chemical knowledge is required to filter out spurious signals and improper assignments. We have developed a simple expert system that takes assignments made by a non-intelligent database system and returns the elements found in the spectrum.

The overall system bears a functional ressemblance to one developed by Trassy and Robin [99] for scanning monochromator systems. Major differences are more advanced software features, wider wavelength coverage, and faster data acquisition. The last two advantages are the result of using FTS as the measurement system.

7-1. Defining and finding peaks in atomic emission spectra

The major difficulty in automating spectral interpretation is in designing an algorithm that enables a computer to accurately find peaks in a spectrum. This is a simple task when the peaks are sufficiently wide to include many data points as in chromatography and molecular spectroscopy, where derivative and local maxima methods are used with suitable filtering to locate peaks.

In atomic spectroscopy, the peaks are very narrow and are difficult to distinguish from noise because they essentially share the same frequency components. Because atomic lines appear noise-like, the above methods are not efficient. Only the most intense lines are found, the rest are rejected as noise.

The definition of a peak that we have elected to use is a familiar one, a peak must be distinguishable from the baseline noise of the spectrum. This is the operational definition that a person uses when visually inspecting an atomic spectrum. Teaching this to a machine is the crux of the problem.

In order to distinguish something from the baseline, one must first define what constitutes a normal baseline and measure its noise. The simplest way to accomplish this is to perform a blank measurement and calculate the standard deviation of all the spectral points. The disadvantage of this method is that it requires an additional measurement and the blank must not contain any spectral features.

Two common solutions to this dilemma have the operator either select a typical baseline region for the program to use or directly enter a threshold value for finding peaks. These semi-automated approaches require user intervention but require no assumptions or special knowledge on the part of the program, the user supplying all of the intelligence. It is for this reason that they are unsatisfactory. The approach we use (see Figure 47) is an "intelligent" approach that is fully automatic but is based on certain characteristics of atomic spectra.

The spectrum is broken up into many sections and the mean and standard deviation of each of these sections are calculated and tabulated. The idea is to sample the spectrum so that some samples contain only baseline while others contain some peaks. It is assumed at this point that there are more samples containing only baseline, a reasonable assumption for high resolution atomic spectra in the ultraviolet.

The tabulated values occupy a range with the samples which contain peaks having abnormally high values of the mean and standard deviation, due to the contribution of the peaks. The estimate of the mean and standard deviation of the baseline can now be obtained by taking the median of the values calculated from the samples. The median is used instead of the mean because the mean would include contributions from those samples containing peaks thereby skewing the value. This approach is statistical and requires no knowledge of the actual structure of the spectrum.

Once the mean and standard deviation of the baseline are known a cutoff threshold T can be calculated that discriminates between signal and noise.

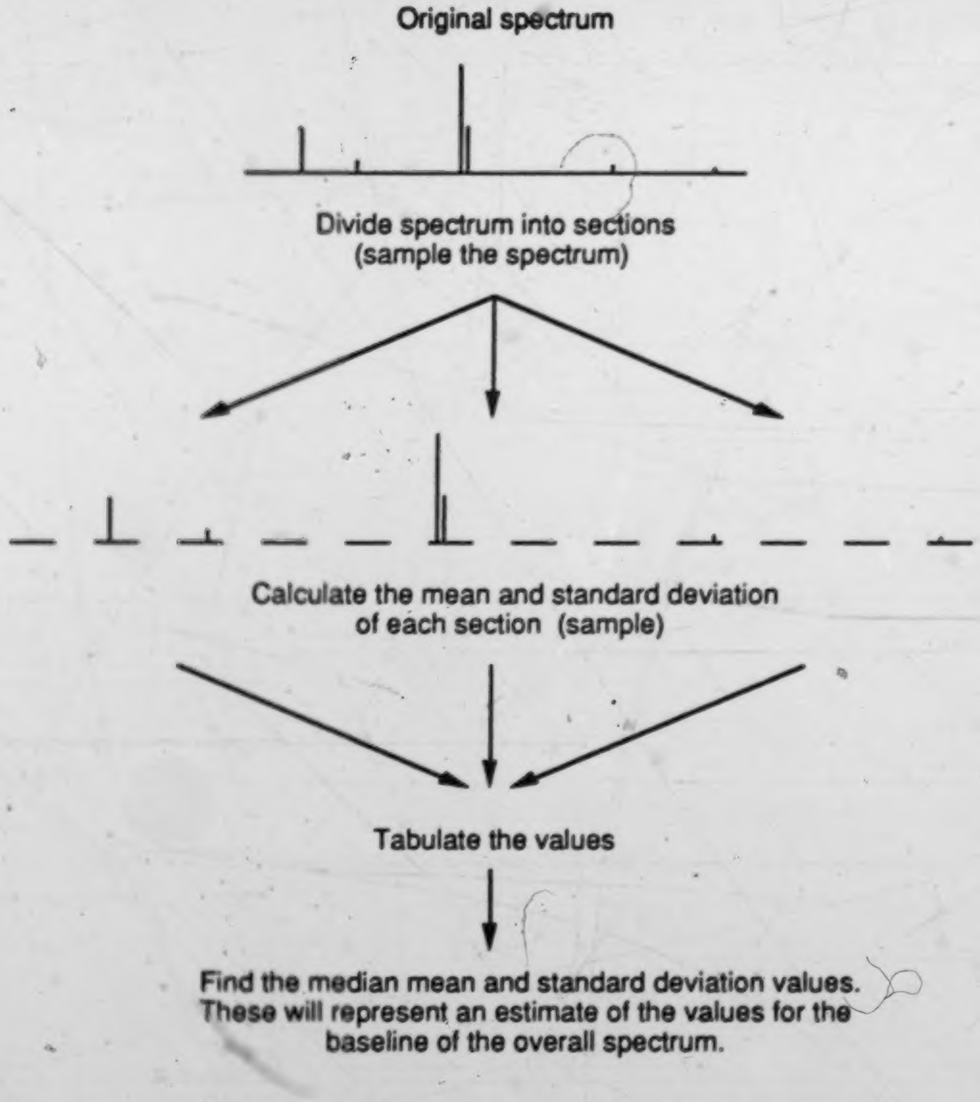

Original spectrum
Divide spectrum into sections
(sample the spectrum)
Calculate the mean and standard deviation
of each section (sample)
Tabulate the values
Find the median mean and standard deviation values.
These will represent an estimate of the values for the
baseline of the overall spectrum.

Figure 47. Procedure for estimating the baseline mean and standard deviation of a spectrum.

$T = m + K s$ where m = mean of baseline
s = standard deviation of baseline
K = a constant

Equation 7-1

Values above the threshold are considered to be signal while those below are noise. Where there are consecutive values above the threshold the position of the maximum is considered to be the location of the peak. The constant K is chosen to minimize the possibility of a false positive and is quite often 3 because a 3σ threshold yields a <1% probability of a false positive for normally distributed noise.

Statistical methods are based on multiple measurements. For a single spectrum we feel that K=5 yields a more realistic threshold since lower values tend to give too many false positives. This result is supported by both simulations and actual measurements of ICP spectra.

In order to help determine a cut-off threshold for the presence of a peak in a single spectrum, a series of simulated spectra were generated. Each simulated spectrum was composed of the sum of a noiseless spectrum and a noisy baseline. The simulated spectrum consisted of 128 points with four single point peaks. For repeated spectra, only the noisy baseline was changed.

Two types of noise were simulated, a uniform distribution and a gaussian distribution. The peaks have values corresponding to 3, 33, 4, and 5 times the standard deviation of the background and are located at points 25, 40, 50, and 70 respectively. Two samples of each type of noisy spectra are shown in Figures 48 and 49.

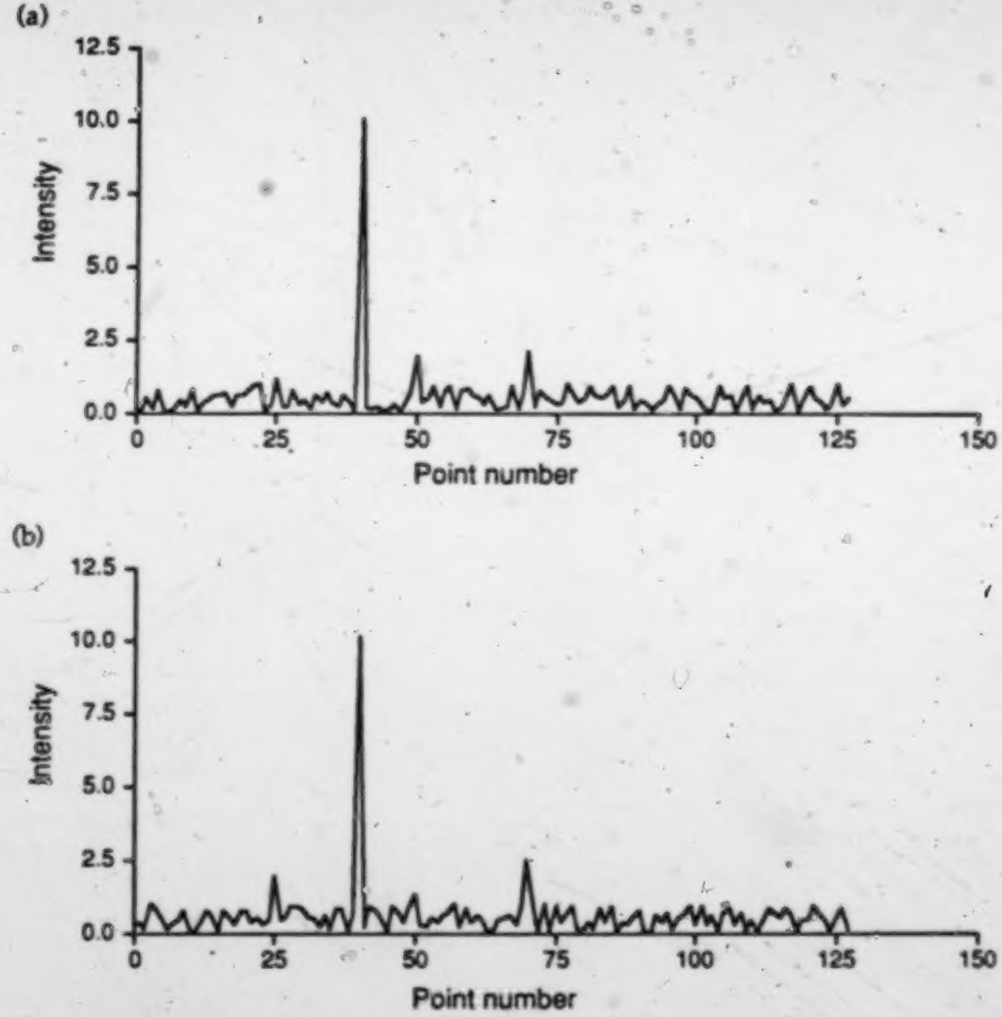


Figure 48. Two spectral simulations using uniformly distributed noise. Peaks that are 3, 33, 4, and 5 times the standard deviation of the baseline are located at points 25, 40, 50, and 70; respectively.

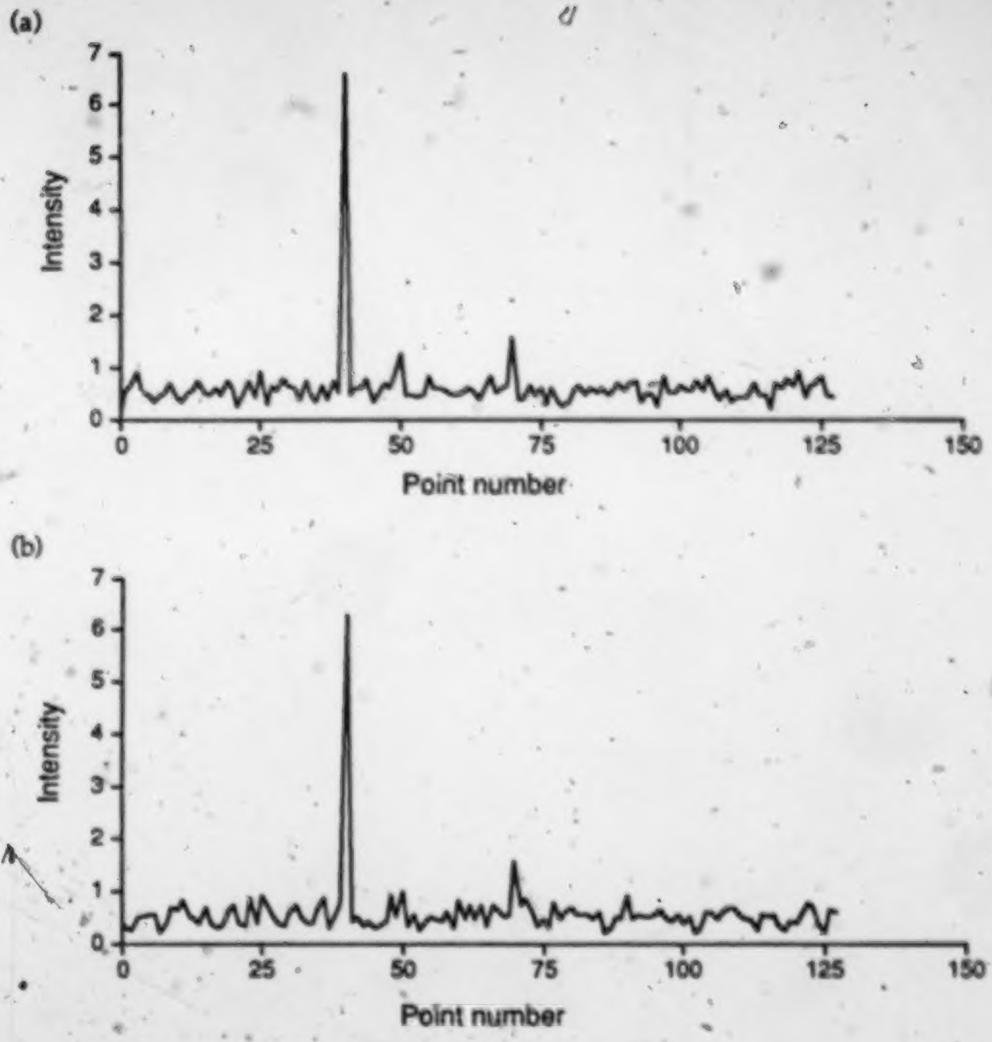


Figure 49. Two spectral simulations using Gaussian distributed noise. Peaks that are 3, 33, 4, and 5 times the standard deviation of the baseline are located at points 25, 40, 50, and 70, respectively.

From a visual inspection of the four spectra the peaks that correspond to 3 and 4 standard deviations are not reliably discernible as signals distinct from the baseline. Sometimes they are apparent and at other times they are not. Only the 5 standard deviation peak is consistently distinct from the baseline in all the spectra.

The noise distribution in an actual ICP spectrum is shown in Figure 50 by a frequency histogram of a region of baseline. The region consists of 201 points (280.7263 – 279.2143 nm) from a water blank spectrum calculated from 32 16K interferograms sampled with the HeNe clock, i.e. the spectrum is aliased. The horizontal axis has been transformed to show the distribution as a function of separation from the mean in units of standard deviation.

The distribution of values about the mean is not quite symmetric, there is a tendency to tail towards higher intensities. In fact, the distribution appears more Poisson than Gaussian which is not surprising since the spectrum is photon noise limited. What is also important is the presence of significant numbers of values at or above the 3 standard deviation limit normally used as the threshold of quantitation. This distribution is the basis for our choice of 5 standard deviations as a more reliable threshold of quantitation when dealing with individual spectra.

7-2. Assigning peaks in atomic emission spectra

Once a peak is discovered, its wavelength is calculated. If the spectrum is aliased then the corresponding wavelengths in each aliasing region within the bandpass of the instrument are calculated. The peak is then tentatively assigned simply by wavelength matching with a database of known lines.

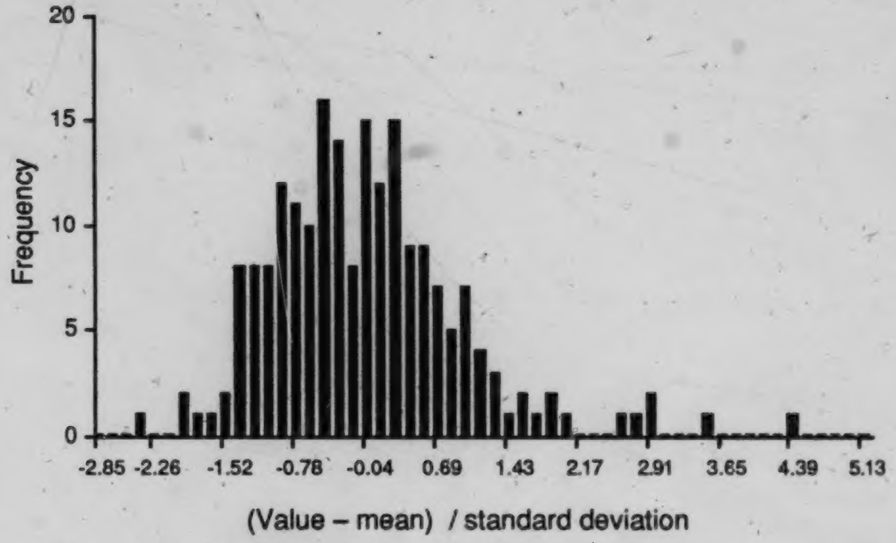


Figure 50. A frequency histogram for a baseline region in a spectrum of 10 mg/L Mg. The values are scaled to show the relative distance from the mean baseline value in units of standard deviation.

The main database is currently managed offline using a commercial database manager (Reflex Plus from Borland) and a properly ordered subset containing the information required to assign peaks is exported for use by the spectral analysis program. The spectral analysis program is only capable of reading and using a pre-processed database. It contains no database management capabilities. The ASCII database exported by the database manager is further processed to convert it into a binary memory image and to create a hash table index. This is done to speed up reading the database into memory and to speed up searches.

Each database entry that matches the peak wavelength is appended to a list of possible assignments for that peak. If a peak has more than one wavelength because of aliasing, the database is searched for each wavelength and each match is combined into a single list of possible assignments. In order to minimize search time only the region of the database near each wavelength is searched instead of the entire database. To enable this, a hash table index to the database is maintained.

An observed wavelength is considered to match a database entry if the difference in wavelength is less than a pre-determined tolerance.

$$|\lambda_{observed} - \lambda_{database}| \leq t \qquad \text{Equation 7-2}$$

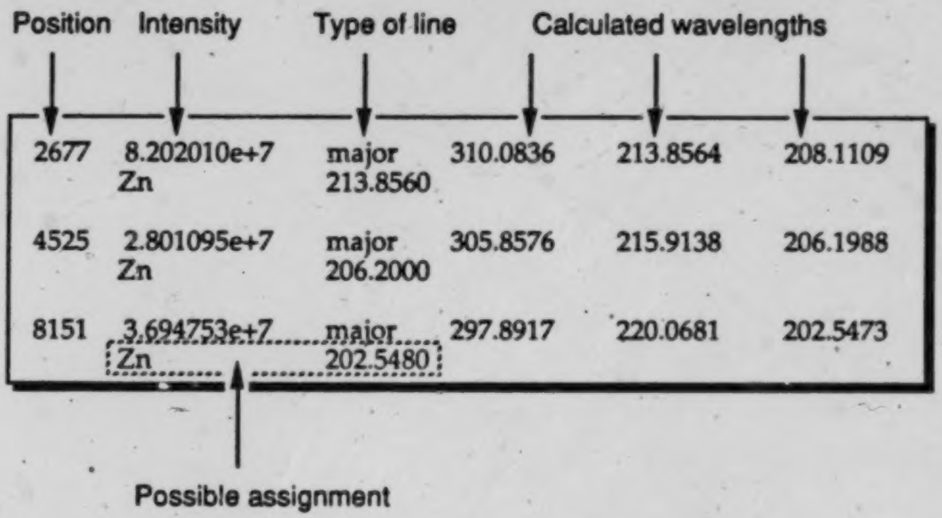
By allowing the two values to differ slightly, the effect of small errors in either value is minimized. This tolerance is normally set to the resolution of the spectrum, which ensures that possible spectral overlaps will also be considered.

The result of finding and assigning lines in a Zn spectrum is shown in Figure 51. The spectrum was calculated from a 64K interferogram sampled with the basic HeNe clock and a solar blind PMT was used as the detector. We have limited the number of aliases to three, the fourth, fifth, and sixth because these are the only ones possible with a solar blind detector.

The peak detection algorithm correctly found the only three spectral features in this spectrum and correctly assigned the lines in spite of aliasing. The algorithm actually used has two levels of discrimination, 5 standard deviations to find all peaks and 10 standard deviations for major features only. The higher threshold is useful in rejecting spurious peaks due to modulation effects on the interferogram and is the setting used for the example shown in Figure 51.

In the case of the Zn spectrum, setting the threshold to 5 standard deviations results in many more spectral features being found (Figure 52). However, direct visual inspection of the spectrum indicates that they are most likely modulation components of the three Zn lines. There is a tendency for them to be found on the wings of an intense line or approximately 90 or 1600 points away (27 or 488 Hz).

The wavelength accuracy of ICP-FTS and the ability of our software to take spectral resolution into account allows the program to correctly assign observed lines using a database of literature wavelengths. Thus any database using literature values can be used with the program.


Position
Intensity
Type of line
Calculated wavelengths
Possible assignment

```
2677   8.202010e+7     major    310.0836     213.8564     208.1109
       Zn              213.8560

4525   2.801095e+7     major    305.8576     215.9138     206.1988
       Zn              206.2000

8151   3.694753e+7     major    297.8917     220.0681     202.5473
       Zn              202.5480
```

Figure 51. Typical results from a spectral search of a aliased Zn ICP spectrum with tentative matches from a database of literature values.

469	252594.1	minor	315.2886	211.4489	210.4425
1096	283163.8	minor	313.7929	212.1270	209.7751
2478	266704.0	minor	310.5457	213.6371	208.3189
2486	235237.5	minor	310.5271	213.6459	208.3105
2491	235036.8	minor	310.5155	213.6514	208.3053
2649	241705.2	minor	310.1486	213.8255	208.1401
2654	233110.7	minor	310.1370	213.8310	208.1349
2660	250306.5	minor	310.1231	213.8376	208.1286
2677	8.202010e+7 Zn	major 213.8560	310.0836	213.8564	208.1109
2827	380438.7	minor	309.7363	214.0219	207.9544
2866	340873.8	minor	309.6461	214.0650	207.9137
2869	251145.4 W	minor 207.9110	309.6391	214.0683	207.9106
4525	2.801095e+7 Zn	major 206.2000	305.8576	215.9138	206.1988
4715	282468.1	minor	305.4297	216.1276	206.0042
6498	236168.5	minor	301.4711	218.1546	204.1957
8151	3.694753e+7 Zn	major 202.5480	297.8917	220.0681	202.5473
8293	235055.2	minor	297.5882	220.2341	202.4069
8299	248635.1	minor	297.5754	220.2411	202.4010

Figure 52. Peaks from a Zn ICP spectrum showing minor spectral features in addition to the three Zn lines.

7-3. A rule-based expert system for identifying elements from atomic emission spectra.

The above software contains no chemical knowledge. Lines are assigned by matching wavelengths without any regard for spectral overlaps or whether the assignment makes chemical sense. An example of this is shown in Figure 52 where a modulation component has been assigned to tungsten. If tungsten was truly present, then its other major lines should also be present. This is a situation similar to the DC arc experiment in the Chemistry 517 course offered by the Department of Chemistry at the University of Alberta. Many students performing the qualitative analysis portion of the experiment assign a line to americium and then report its presence.

In order to resolve the tentative line assignments made by the spectral search software, an expert system is used to apply a little chemical knowledge to filter the results of the preliminary search. The basic strategy is search for the major lines of each element that has been tentatively detected in to the sample spectrum. This is similar to what a chemist would do if the sample is suspected to contain Mg. The chemist would search the spectrum for the 279.55 nm, 280.27 nm, and 285.21 nm lines to confirm its presence. Currently only the wavelengths of the major lines are used but a more sophisticated system would also attempt to match the known intensity ratios.

I decided to write the expert system in Prolog because it offers flexibility and fast development time and is more database oriented than Lisp. The heart of our expert system are the rules used for determining the presence of each particular element. The bulk of the program is involved with reading and parsing the input data into a form that can be conveniently processed.

The result of the spectral search is read into an internal Prolog list structure where each item in the list corresponds to a single line and its assigned elements. Unassigned lines are discarded to save memory and to speed processing. Each item in this list consists of five components, the position of the line, the intensity, the type of line, a list of wavelengths, and a list of tentative assignments. Each item in the list of tentative assignments contains the symbol for an element and the wavelength of the line that matches one of the wavelengths for the observed line.

The elements are extracted from the list of tentative assignments for each line and are collated and sorted to arrive at a list of possible elements contained in the sample from which the spectrum was obtained. The expert system then searches the list of observed lines for those that match the principal lines of each suspected element.

The principal lines for each element are stored in rules. Each rule is a Prolog goal and there can be more than one rule for each element. A typical rule is shown in Figure 53. The rules are kept in a separate file so that they can be updated without affecting the main program and so that different sets of rules can be tried.

In order to establish the presence of a particular element, the program searches the list of lines for the three strongest lines for the given element. If all three are found then the presence of the element is positively established and the goal returns "true". However, if only the two strongest lines are found then it is possible that the element is present, but in smaller quantities. In this case the goal returns "maybe". If these conditions cannot be met and

```
element('Mg', L, X) :-
      find(L, ['Mg', 279.5531]),
      find(L, ['Mg', 280.2710]),
      (find(L, ['Mg', 285.2129]) -> X=true ; X=maybe).
```

Figure 53. A rule for determining the presence of Mg.

all rules for the element fail then the result is assumed to be "false", either no lines were found or only one, in which case the evidence is inconclusive.

The order in which the lines are searched is important. If a weak line is present but not one of the stronger ones then it is unlikely that the element is present. Because the relative line intensities must be known in order to formulate the rules, the rules become source specific. In the case of the ICP, because the intensity ratios can vary with operating conditions, the rules are specific to a particular set of conditions.

7-4. The determination of major elements in brass, steel, and stainless steel.

The software was tested by using it to find the major elements in some real samples. The ones chosen were the NBS stainless steel SRM 121d, the NBS low alloy steel SRM 362, and the NBS Naval brass B SRM 1107. The certified compositions of these samples are listed in Tables 12 to 14. The rules used by the expert system were derived from our own ICP-FTS measurements of 22 elements.

The steel and stainless steel samples were dissolved using a modification of the method described by Fernando [100]. The procedure is simple and was used by another member of our group for trace analysis with good results [101].

A 1 g sample is weighed in a plastic beaker and 2 ml of concentrated HNO_3 is slowly added. This is followed by the slow addition of 4 ml of concentrated HCl. The mixture is then warmed before addition of 2 ml of HF and is kept warm until the reaction subsides. The contents of the beaker is then transferred to the Teflon cup of a Parr bomb and heated.

Element	Certified composition (nominal weight percent)
C	0.067
Mn	1.80
P	0.019
S	0.013
Si	0.54
Cu	0.121
Ni	11.17
Cr	17.4
Mo	0.165
Co	0.10
Ti	0.342

Table 12. Certified composition of NBS stainless steel SRM 121d.

Element	Certified composition (nominal weight percent)
C	0.160
Mn	1.04
P	0.041
S	0.036
Si	0.39
Cu	0.50
Ni	0.59
Cr	0.30
V	0.040
Mo	0.068
W	0.20
Co	0.30
Ti	0.084
As	0.09
Sn	0.016
Al	0.09
Nb	0.29
Ta	0.20
Zr	0.19
Ca	0.0002
B	0.0025
Pb	0.0004
Sb	0.013
Ag	0.0011
Ce	0.0019
Nd	0.0007
Mg	0.0006

Table 13. Certified composition of NBS low alloy steel SRM 362.

Element	Certified composition (nominal weight percent)
Cu	61.21
Zn	37.34
Pb	0.18
Fe	0.037
Sn	1.04
Ni	0.098

Table 14. Certified composition of NBS Naval brass B SRM 1107.

My procedure had to deviate from Fernando's because the only available oven which could accommodate our bomb was already in use and was set at 160°C instead of the recommended 180°C. I heated our samples at 160°C for two hours (twice that recommended at 180°C) and obtained complete dissolution. The resulting mixture was cooled and diluted to 100 ml with de-ionized water.

Preparation of the brass sample was much simpler, a 0.5 g sample was dissolved in 5 ml of warm concentrated HNO_3 and then diluted to 100 ml. All acids used were reagent grade.

Each of the dissolved samples was further diluted 1:10 prior to analysis to give solutions that were approximately 1000 mg/L for steel and 500 mg/L for brass.

The ICP operating parameters were 1.5 kW forward power, 13.3 lpm (outer), 1.0 lpm (intermediate), and 0.8 lpm (central) gas flows. The plasma was observed at 15 mm above the load coil with our Michelson interferometer using an R166 solar blind PMT as a detector. 64K interferograms were obtained using both the basic HeNe clock (aliased) and a 4× clock (de-aliased). Each final interferogram consisted of 64 co-added scans and was transformed without further processing.

The software was then instructed to find all spectral lines in the resulting spectra (see Figures 54 to 56) and to make tentative assignments from a database of major lines (relative intensity >0.1) for the elements C, Cd, Co, Cr, Fe, Hg, La, Mg, Mn, Mo, Ni, Pb, Si, Sn, Ta, Th, U, V, W, Zr, and Zn. The software was set to calculate the baseline standard deviation using 60

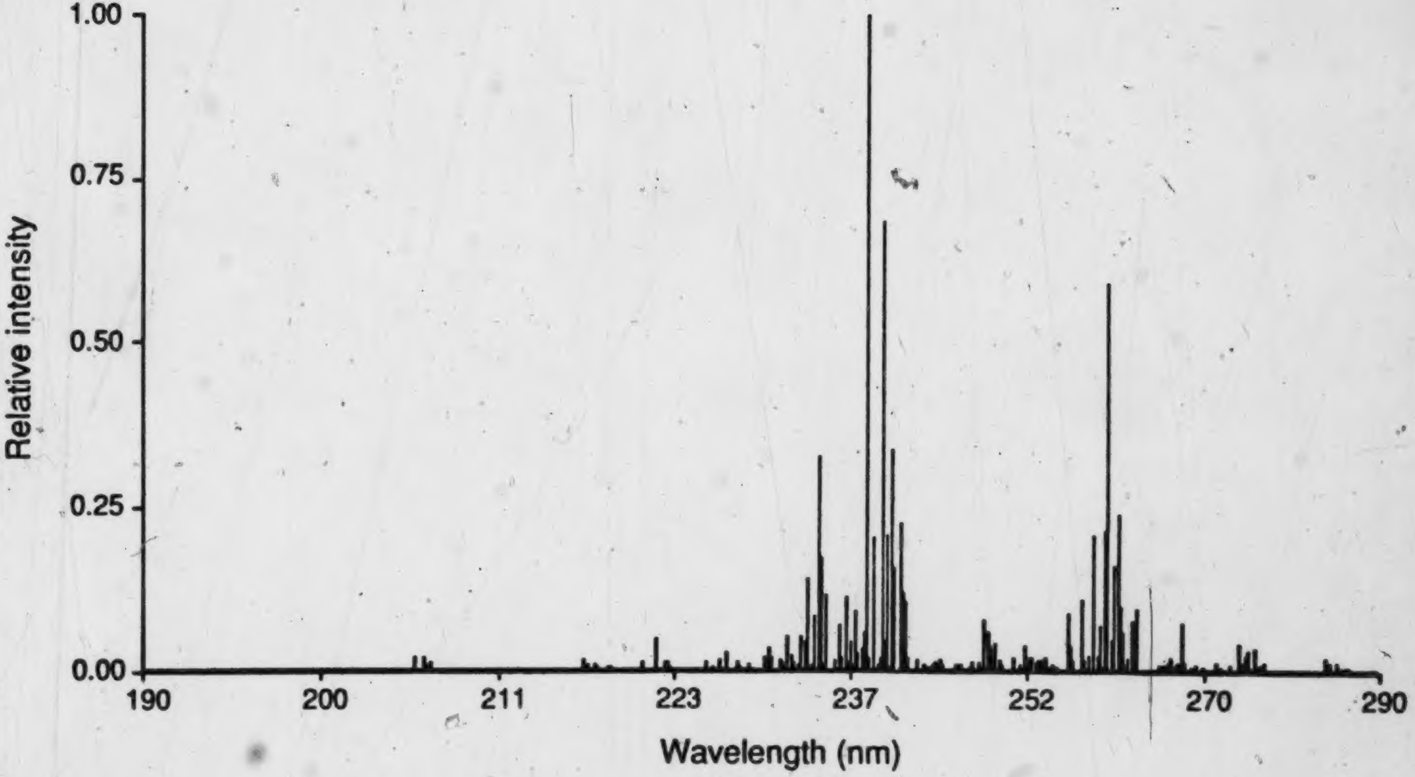


Figure 54. The ICP ultraviolet spectrum of NBS stainless steel SRM 121d.

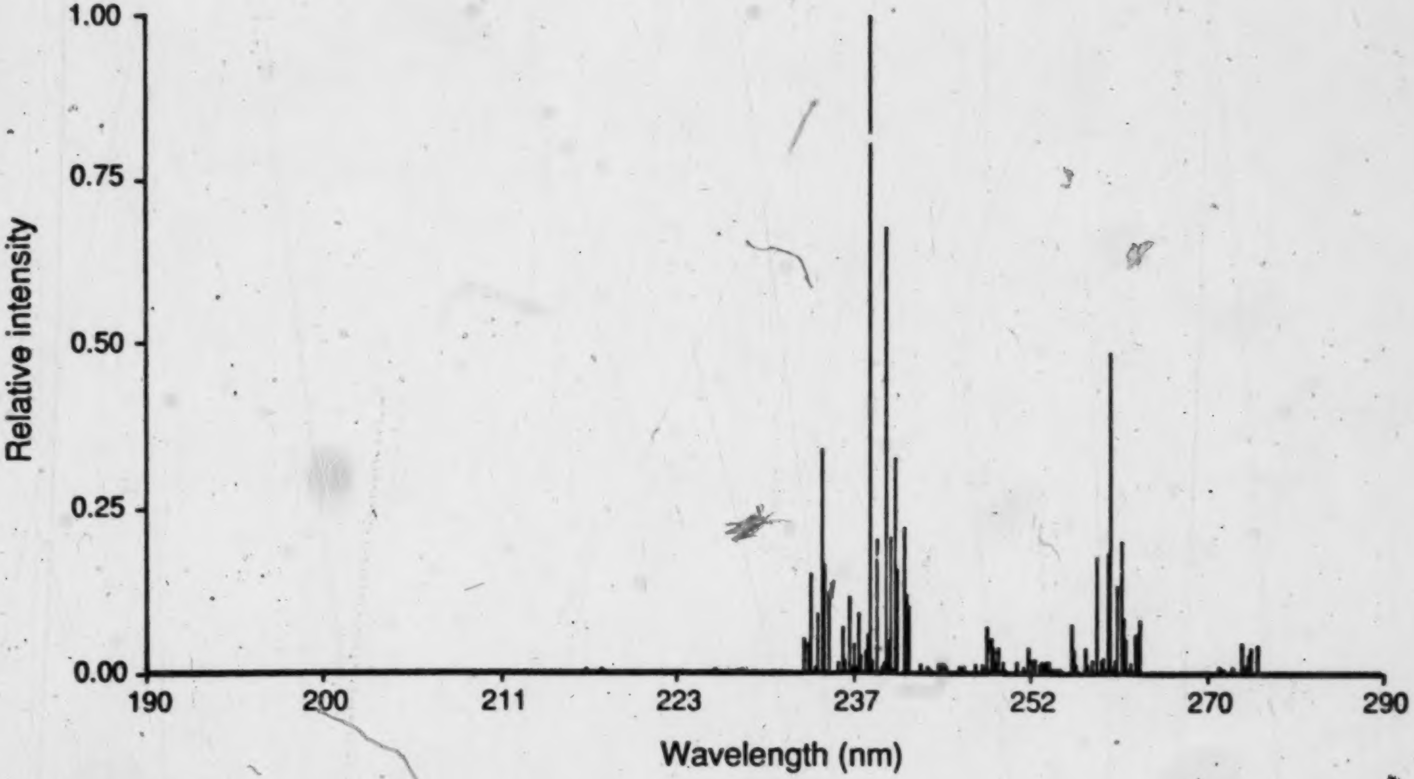


Figure 55. The ICP ultraviolet spectrum of NBS low alloy steel SRM 362.

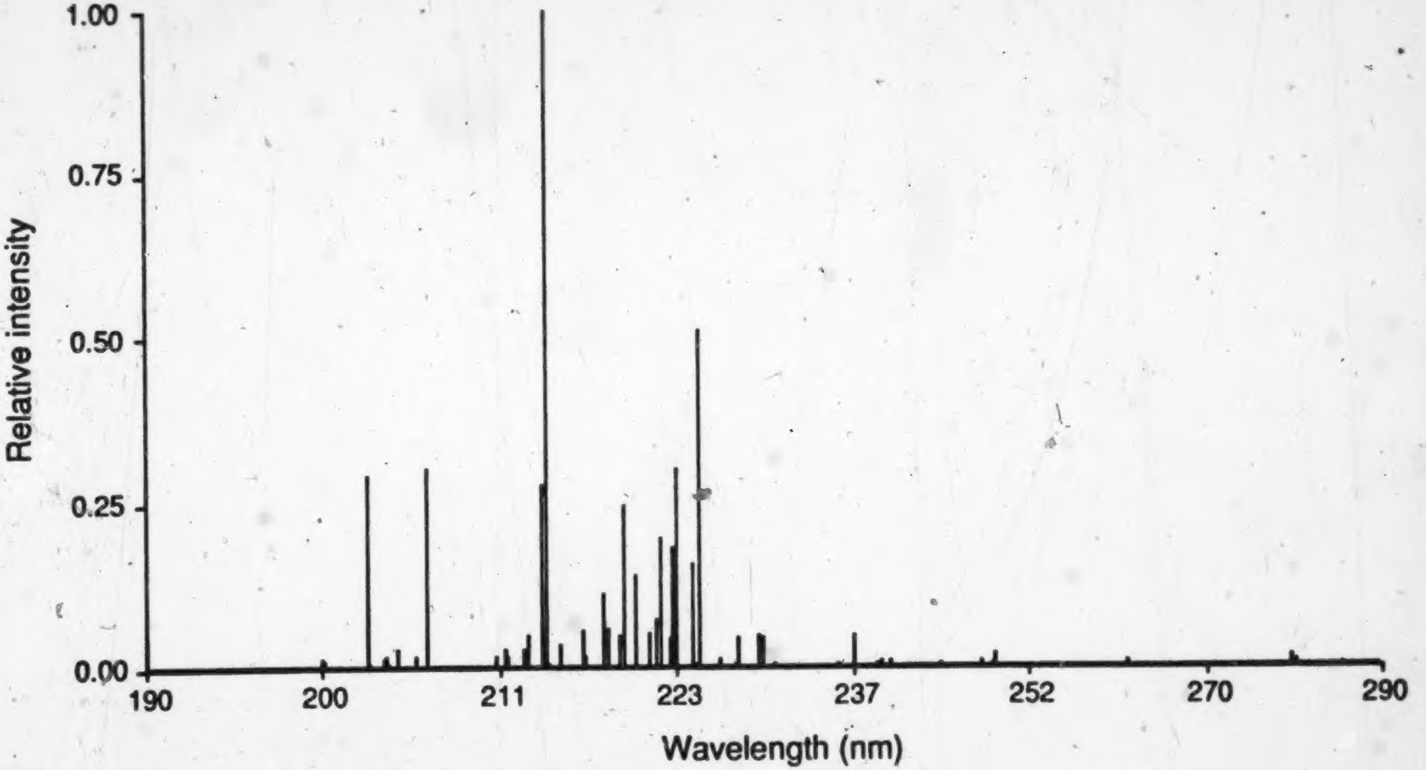


Figure 56. The ICP ultraviolet spectrum of NBS brass SRM 1107.

point samples, to report all peaks greater than 5 standard deviations, and to attempt to match lines within ±2 spectral points of each peak. The resulting line lists were then processed by the expert system to determine the elements that were present.

The lines that were assigned in the de-aliased spectrum of stainless steel (SRM 121d) are listed Table 15. Most lines were correctly assigned to major component elements of the steel but several were mis-assigned to elements not commonly found in steel because a spectral overlap existed. Figures 57 to 59 show an ~2 nm region centered on each of the three strongest lines of Ni, Mn, and Cr. Each line was detected and correctly assigned by software even in the presence of neighbouring Fe lines.

The rules used in the expert system require that the three most intense lines for a given element must be found for the element to be definitely present. If only the two most intense lines are found then the element is labelled as a possible. The line intensities were taken from our database of ICP emission lines. This is important; an alternate source may have different relative intensities causing inappropriate lines to be used in the rulebase.

The expert system correctly found all the major components of the stainless steel sample, however it was unable to find any of the trace elements (<1%) in the sample. An example of the output of the expert system is shown in Figure 60 and a summary of the results for the stainless steel sample is shown in Table 16.

Manual inspection of the spectrum also failed to detect any of the trace components (see Figure 61), with the possible exception of Si where the most intense line may be present (see Figure 61c). In the case of the aliased spec-

#	rel. int.	type	λ	assigned lines			
15097	1.301266e+6	major	205.5620	Cr	205.5590		
15243	1.443799e+6	major	206.1588	Cr	206.1567		
15338	983722.9	major	206.5489	Cr	206.5500		
17658	1.089960e+6	major	216.5576	Ni	216.5542		
17736	560185.2	major	216.9110	Ni	216.9087		
17858	657870.7	major	217.4660	Ni	217.4660		
17869	465760.7	major	217.5162	Ni	217.5139		
18075	478357.5	major	218.4603	Ni	218.4591		
18438	323971.5	minor	220.1440	Ni	220.1394		
18551	706785.9	major	220.6735	Ni	220.6700		
18758	3.441691e+6	major	221.6500	Ni	221.6465		
18894	916220.4	major	222.2963	Ni	222.2959		
18934	1.011635e+6	major	222.4871	Ni	222.4839		
18965	509731.1	major	222.6352	Ni	222.6316		
19533	801071.4	major	225.3841	Ni	225.3841		
19749	1.109173e+6	major	226.4473	Ni	226.4449		
19865	1.807823e+6	major	227.0225	Ni	227.0200	W	227.0225
20037	797170.9	major	227.8807	Ni	227.8757		
20202	354387.5	major	228.7101	Ni	228.7076		
20389	1.000086e+6	major	229.6574	Ni	229.6536		
20400	1.246525e+6	major	229.7134	Ni	229.7134		
20407	636616.5	major	229.7490	Ni	229.7478		
20422	693159.1	major	229.8254	Ni	229.8254		
20515	2.412069e+6	major	230.3002	Ni	230.2989		
20670	1.163279e+6	major	231.0958	Ni	231.0945		
20697	799165.8	major	231.2350	Ni	231.2337		
20769	3.735848e+6	major	231.6069	Ni	231.6050		
20846	1.667181e+6	major	232.0059	Ni	232.0020		
20872	745067.9	major	232.1410	Ni	232.1371		
20956	607506.5	major	232.5784	Ni	232.5784		
21090	9.915614e+6	major	233.2796	Fe	233.2783		
21293	2.312949e+7	major	234.3499	Fe	234.3486		
21384	8.244780e+6	major	234.8329	Fe	234.8290		
21692	7.785502e+6	major	236.4826	Fe	236.4813		
22009	7.056649e+7	major	238.2049	Fe	238.2021		
22129	1.425972e+7	major	238.8634	Fe	238.8620		
22207	281120.2	minor	239.2934	W	239.2920		
22256	4.835574e+7	major	239.5643	Fe	239.5616		
22282	286264.3	minor	239.7083	W	239.7097		
22288	258000.8	minor	239.7416	Co	239.7374		
22321	1.455495e+7	major	239.9247	Fe	239.9233		
22423	2.373926e+7	major	240.4923	Fe	240.4881		
22454	1.106714e+7	major	240.6653	Fe	240.6653		
22523	1.584255e+7	major	241.0514	Fe	241.0514		
24328	284714.2	minor	251.6100	Si	251.6100		
25288	7.523376e+6	major	257.6114	Mn	257.6082		
25440	1.460561e+7	major	258.5880	Fe	258.5864		
25561	4.703532e+6	major	259.3707	Ta	259.3658	Mn	259.3707
25612	578620.1	major	259.7020	Ta	259.7037		
25649	4.154713e+7	major	259.9429	Fe	259.9381		
25745	3.458718e+6	major	260.5701	Mn	260.5668		
25766	1.125928e+7	major	260.7077	Fe	260.7077		
25839	1.686909e+7	major	261.1871	Fe	261.1854		
25869	6.914078e+6	major	261.3846	Fe	261.3813		
25873	246432.9	minor	261.4110	Pb	261.4159		
25926	4.226606e+6	major	261.7608	Fe	261.7608		
26048	5.268332e+6	major	262.5695	Fe	262.5645	Ta	262.5728
26087	4.901860e+6	major	262.8291	Fe	262.8274		
26133	6.666736e+6	major	263.1359	Fe	263.1309		
26607	861728.7	major	266.3399	Cr	266.3399		
26645	1.341807e+6	major	266.6001	Cr	266.6001		
26685	567923.1	major	266.8746	Cr	266.8694		
26730	732148.1	major	267.1840	Cr	267.1789		
26745	593868.9	major	267.2874	Cr	267.2822	Mn	267.2822
26807	5.115995e+6	major	267.7153	Cr	267.7153		
26831	1.012500e+6	major	267.8813	Cr	267.8778		
26950	391521.1	minor	268.7074	Cr	268.7074		
27007	633383.6	major	269.1049	U	269.1014	Cr	269.1032
27489	2.805785e+6	major	273.9542	Fe	273.9524		
27740	816125.8	major	274.3238	Fe	274.3184		
27785	1.692050e+6	major	274.6508	Fe	274.6472		
27792	1.202320e+6	major	274.7018	Fe	274.6983		
27819	490331.9	major	274.8984	Cr	274.8966		
27824	2.217096e+6	major	274.9349	Fe	274.9312		
27843	416929.4	minor	275.0735	Cr	275.0716		
27859	278939.5	minor	275.1903	Cr	275.1848		
27912	2.359842e+6	major	275.5779	Fe	275.5724		
28005	593744.8	major	276.2608	Cr	276.2571		
28058	901717.7	major	276.6515	Cr	276.6515		
28972	1.537749e+6	major	283.5670	Cr	283.5612		
29070	988348.6	major	284.3291	Cr	284.3232		
29134	794217.4	major	284.9855	Cr	284.9816		
29228	423584.5	minor	285.5664	Cr	285.5664		
29316	402029.6	minor	286.2602	Cr	286.2542		
29348	386518.9	minor	286.5133	Cr	286.5094		
29368	281709.8	minor	286.6718	Cr	286.6718		

Table 15. Software assigned lines for SRM 121d.

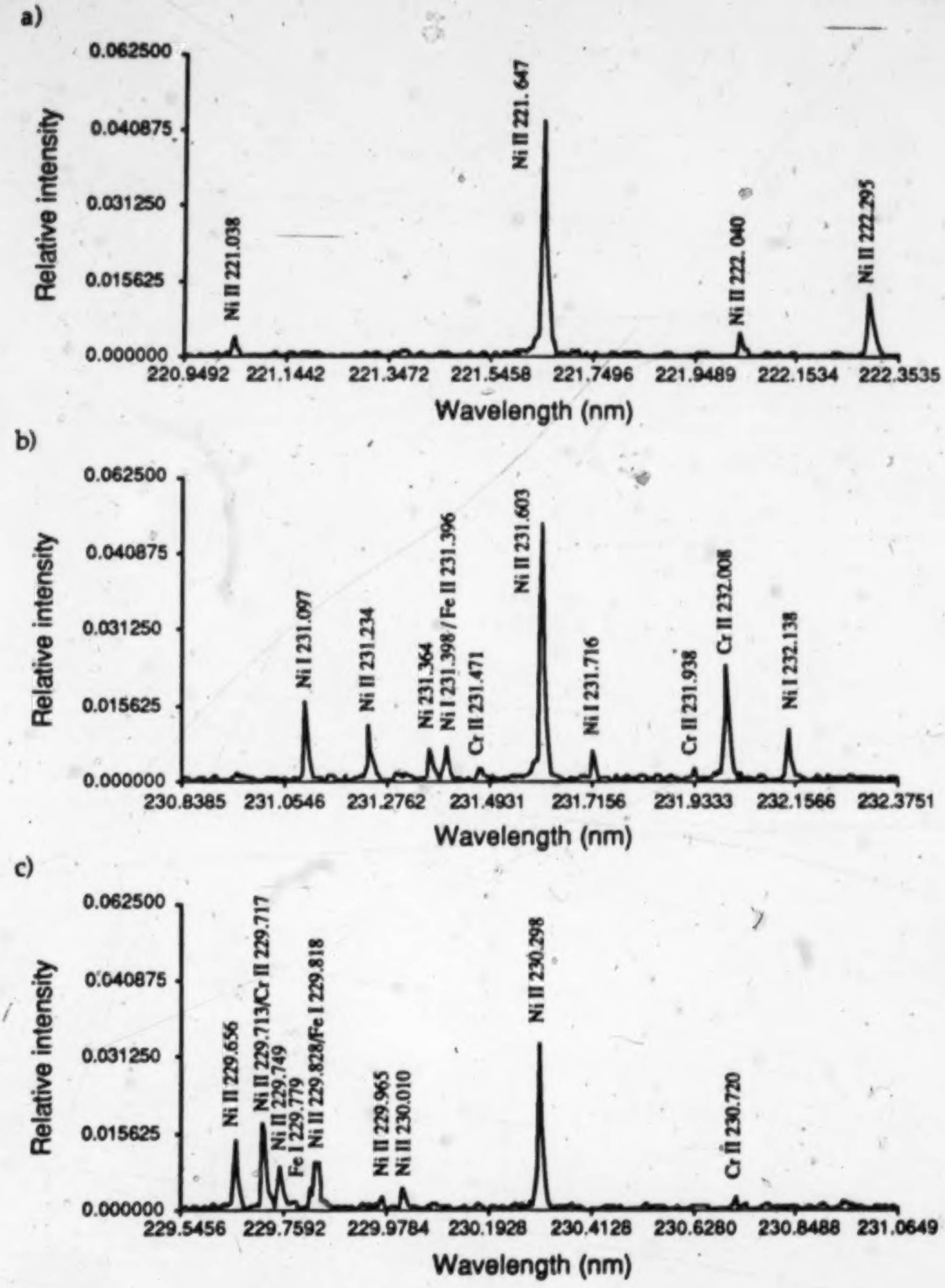


Figure 57. Expanded views of the regions ~1 nm on either side of the three principle Ni lines in the spectrum of SRM 121d.

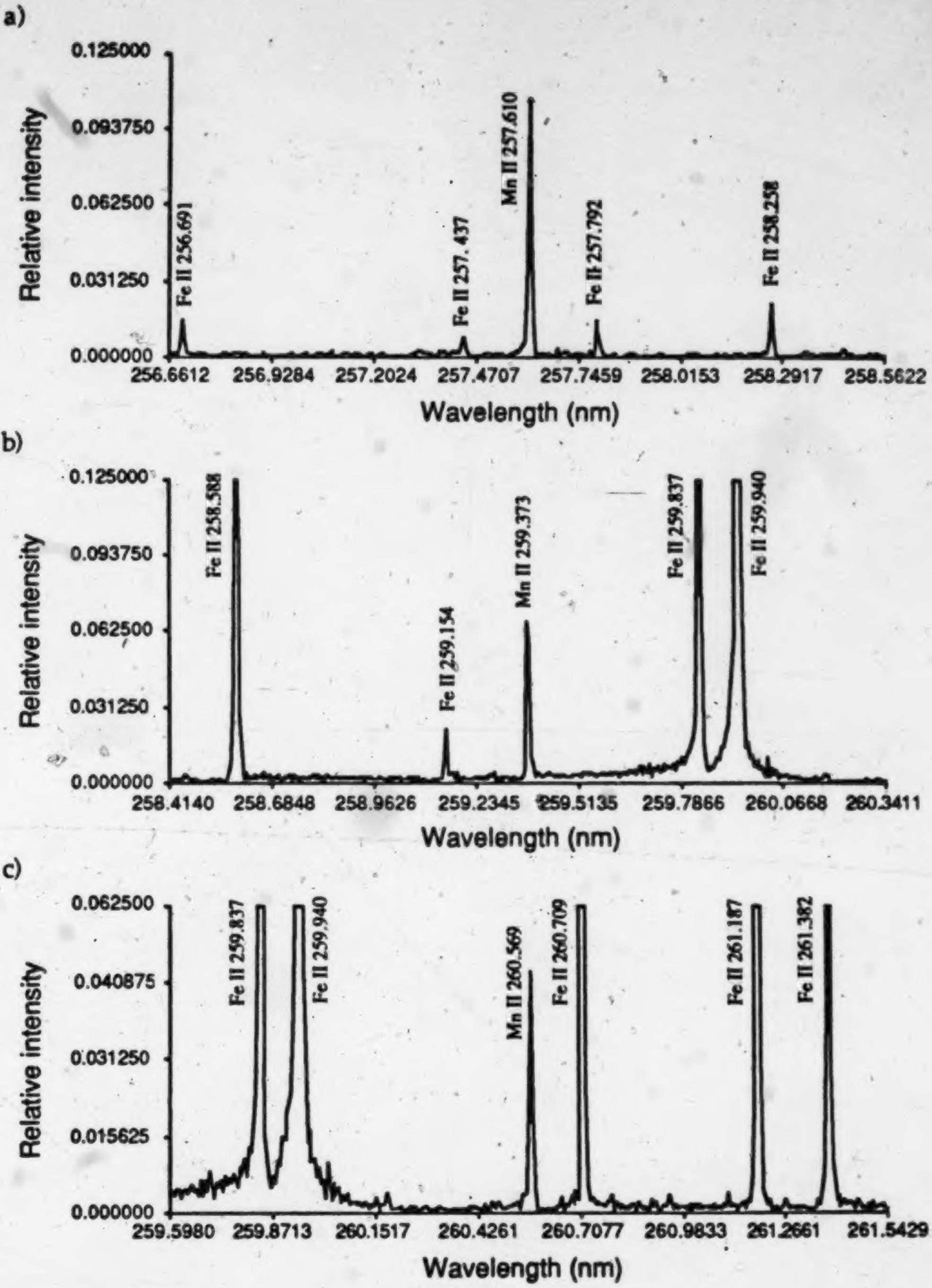


Figure 58. Expanded views of the regions ~1 nm on either side of the three principle Mn lines in the spectrum of SRM 121d.

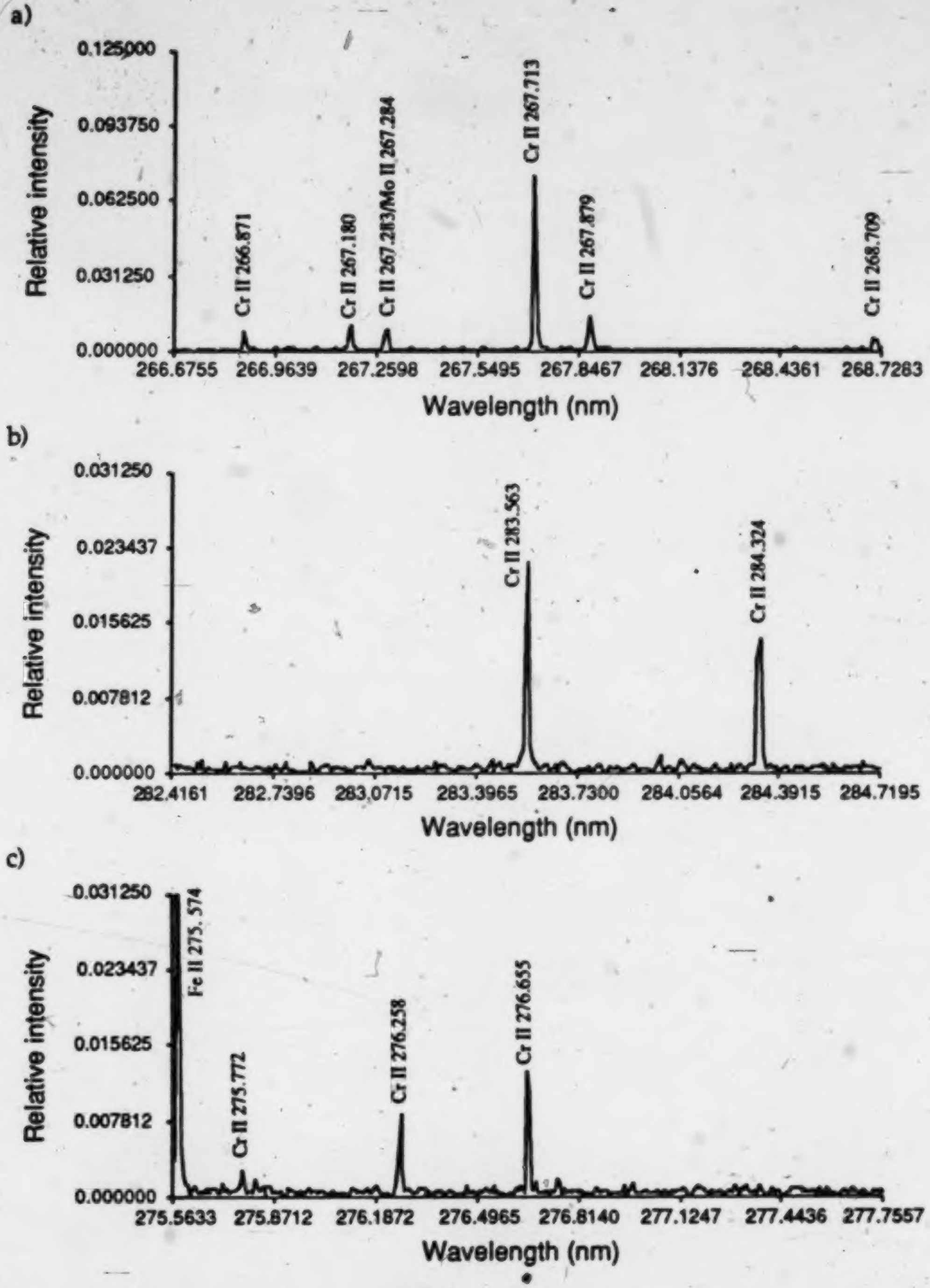
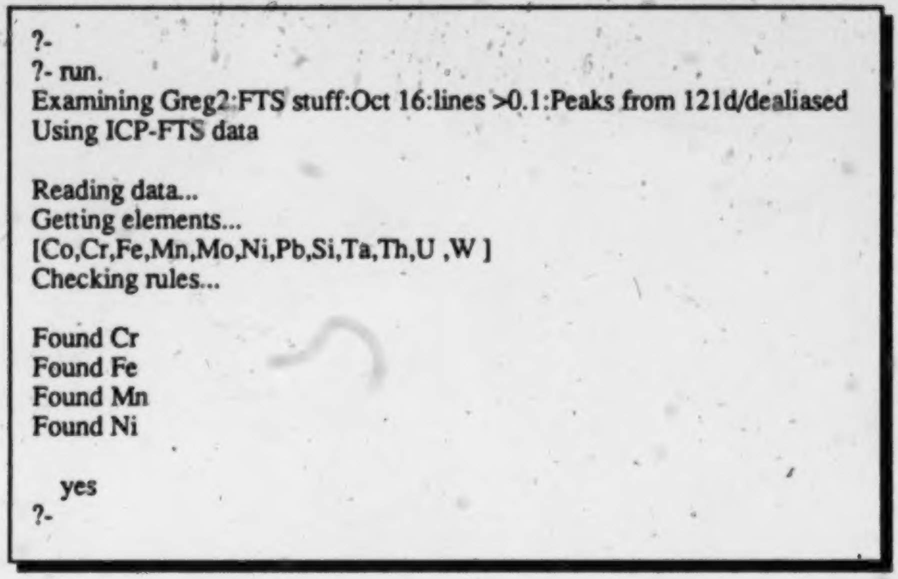


Figure 59. Expanded views of the regions ~1 nm on either side of the three principle Cr lines in the spectrum of SRM 121d.



```
?-
?- run.
Examining Greg2:FTS stuff:Oct 16:lines >0.1:Peaks from 121d/dealiased
Using ICP-FTS data

Reading data...
Getting elements...
[Co,Cr,Fe,Mn,Mo,Ni,Pb,Si,Ta,Th,U ,W ]
Checking rules...

Found Cr
Found Fe
Found Mn
Found Ni

   yes
?-
```

Figure 60.   Output from expert system.

element (% composition)	search result de-aliased	 aliased
Cr (17.4)	found	possible
Ni (11.17)	found	found
Mn (1.80)	found	found
others (<1)	none found	not found

Table 16. Result of software analysis of SRM 121d spectra.

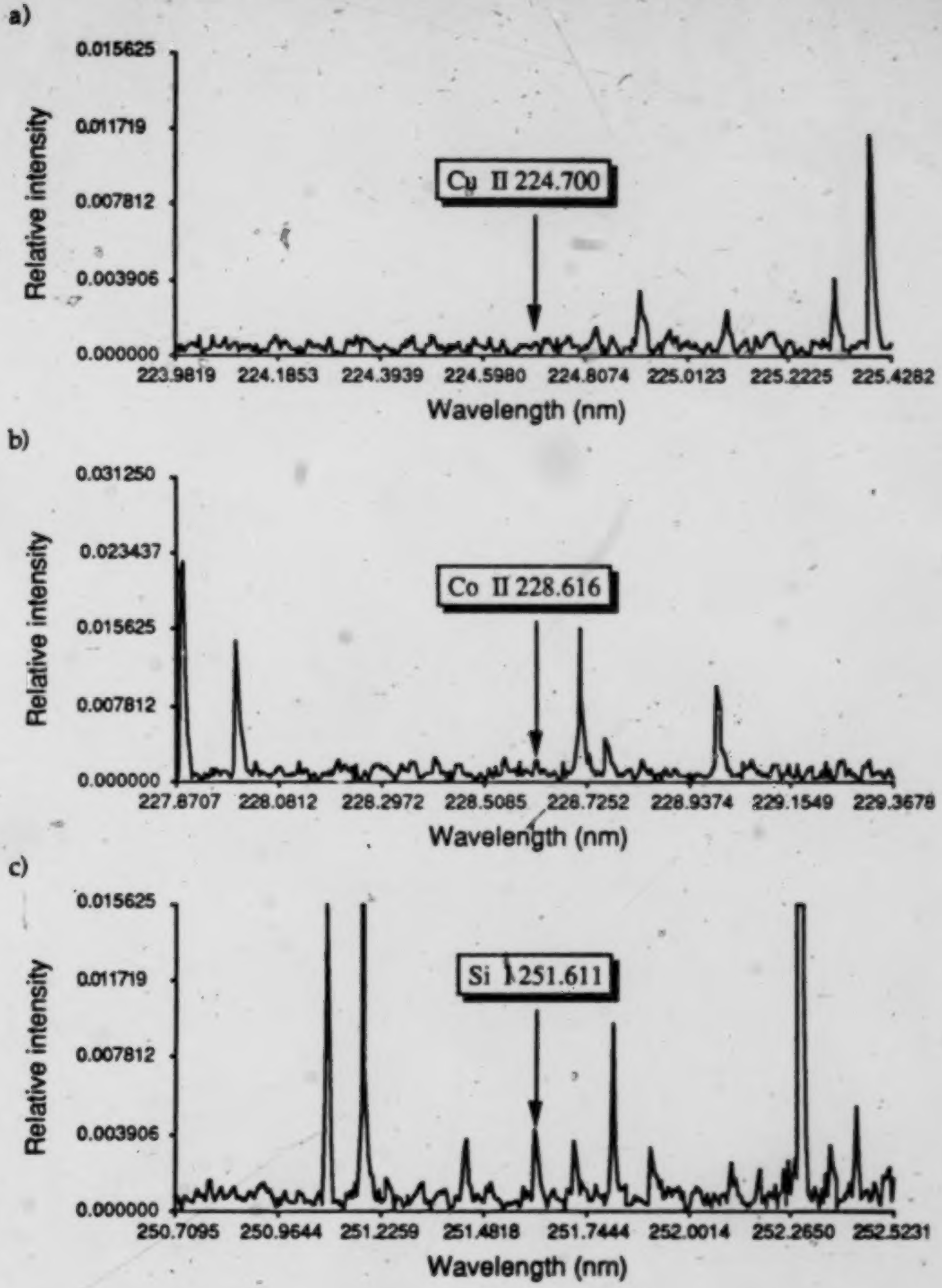


Figure 61. Scale expanded views of the spectrum of SRM 121d where principle lines of Cu (a), Co (b), and Si (c) would be found.

trum the increased background noise obscured the least intense of the three Cr lines used for identification so the software only reported the possible presence of Cr.

Similar results were obtained for both the low alloy steel and the brass samples, major species were correctly identified but trace components were below the detection limit for ICP-FTS. In the low alloy steel only Mn was present at a level high enough to be seen using ICP-FTS (1.04%), however the brass sample generated a surprise.

In addition to Cu and Zn, the prime constituents of brass, the software also detected the possible presence of Mg, which is not supposed to be present in this particular brass sample. It is highly likely that the Mg is a contaminant introduced into the sample solution via tap water residues in the bottom of the plastic bottle used to store the solution. The intensity of the Mg lines are consistent with this this conclusion.

This illustrates one of the major advantages of our automated analysis system. It is capable of detecting and identifying contaminants without any advance knowledge of their existance. This makes the method particularly attractive for quality control applications.

7-5. Conclusions

The qualitative analysis of steels using ICP-FTS shows that I have developed an expert system to automatically perform chemical analyses on unknown samples. The software was able to find and correctly identify the atomic emission lines present in the spectra. That the system was unable to find the trace components is not a fault of the software but is due to the adverse effect of the multiplex disadvantage on detection limits in ICP–FTS.

The current software system is just the starting point for implementing fully automatic, intelligent analysis systems. The goal is not to list out measured values for pre-determined species as is currently done in commercial instrumentation. It is also not to determine which species are present (implemented) and their concentrations (not yet implemented) without *a priori* knowledge of the sample.

The goal is to answer some real world, non-chemical question that requires a non-chemical answer that is based on chemical knowledge. This type of goal was first proposed by Karanassios [102] for a photodiode array based instrument. To make an analogy with automobiles, the goal is to convert analytical instrumentation from stick-shift to automatic transmission.

An example of a real world problem is wear metals in lubricating oil. When a sample of lubricating oil from a gas turbine is analyzed for wear metals the real question is not how much metal is present, but rather, does the unit have to be shutdown in order to change the bearings? This is a non-trivial question since shutting down a turbine, or jetliner, is an expensive proposition, but the consequences of not changing the bearings can be disastrous.

By combining an expert system for trace metals analysis with an expert system for gas turbines it is possible to acquire such answers from an automated system. Each expert system works within its own domain and has no knowledge outside of it. By keeping the software highly modular in this manner it is possible to reconfigure the overall system with a minimum of overhead. For instance, if the gas turbine expert system is replaced with a steel making expert system then the system can be used to control steel production in an automated plant. The chemical expert system knows nothing about gas turbines or steel making as this knowledge isn't necessary for its proper function.

The software that we have written is functional but currently does not address any real world problems. It was written to demonstrate its feasibility and as such, it isn't as robust as a real production system needs to be.

Our method of determining the background noise makes two major assumptions, 1) that much of the spectrum is background and 2) that the noise is constant throughout the spectrum. The first assumption is reasonable for ICP–FTS but the second is not. Because of the multiplex disadvantage in ICP–FTS and the 1/f nature of noise in the ICP, the background noise shows some 1/f variation near strong lines. This generally results in more false positives near strong lines and more false negatives far away in dealiased spectra with only a few strong lines.

In addition, we have observed a general trend for noise to decrease with decreasing wavelength in our spectra. Aliasing and the presence of many strong lines will usually smear out these variations in background noise.

Currently the routine that finds peaks in a spectrum has two fail modes, it is unable to separate peaks that are not baseline resolved and it cannot handle spectra with shifting baselines. In the case of partially resolved peaks only the most intense peak is reported, all others are currently ignored.

This has serious implications in complex multi-element spectra where near spectral overlaps can result in resolvable peaks that are ignored.

The shifting baseline problem is not nearly as critical but it is extremely annoying nonetheless. Currently the threshold is based on the average baseline level, when the actual baseline is lower it is possible that small peaks will be ignored and when the baseline is higher background noise can be shifted above the threshold yielding false positives. The case of the high baseline is the only one that is readily apparent without actual inspection of the spectrum, it is not apparent when peaks are ignored.

The use of intensity ratios in the rule-base of the expert system would add an additional degree of certainty to the identification of constituent elements and also help resolve possible spectral interferences. Combined with an internal standard it should also be possible to perform automated quantitative analysis.

Chapter 8

Multiline detection limits.

Because of the multiplex disadvantage incurred by ICP–FTS, the detection limits of species analyzed using this technique are degraded. Quieter sources can eliminate much of the disadvantage but photon noise imposes a fundamental limit on the amount of noise reduction possible. Bandpass limiting can also result in significant improvements in detection limits but negates the benefits of FTS's simultaneous wide wavelength coverage.

It is possible to take advantage of the additional information contained in a spectrum to reduce the effect of the multiplex disadvantage on the detection limit of a species. It is our proposal to construct calibration curves and calculate detection limits using all the major lines of a species instead of just a single line as normally done. In this way, the use of available information is maximized in order to reduce some of the uncertainty due to noise.

Such a multi-point calibration curve has been used by Spillane *et al.* [103] to improve signal-to-noise ratios and calibration curve linearity in their ICP–FTS data although no detection limits were reported. Layman [104] has also reported using multivariant statistical analysis with multiple atomic emission lines.

8-1. Theory

The detection limit (dl) is defined as the smallest detectable amount of an analyte species and is commonly given by:

$$dl = \frac{3\sigma C}{x} \qquad \text{Equation 8-1}$$

where C is the concentration of analyte that gives a measured response x and σ is the background standard deviation. This definition is depicted graphically in Figure 62. By using 3σ as the threshold for positively identifying a response, the probability of a false positive is less than 1%. This definition is based on normal statistics.

If an analyte produces multiple responses (i.e. multiple spectral lines), they can be combined into a single overall response Y from which a detection limit can also be calculated.

$$Y = \sum_{i=0}^{N} x_i \qquad \text{Equation 8-2}$$

N is the total number of lines used and x_i is the intensity of the i'th line.

By combining Equations 8-1 and 8-2 a multiline detection limit (dl_N) can be defined as:

$$dl_N = \frac{3\sqrt{N}\,\sigma C}{Y} \qquad \text{Equation 8-3}$$

This assumes that the noise in the blank is the same at each line position. If it is not, then the square root of the sum of the variances must be used instead of $\sqrt{N}\sigma$. This is in effect an integration and can result in improvements in the detection limit if the signal accumulates faster than the noise. Improve-

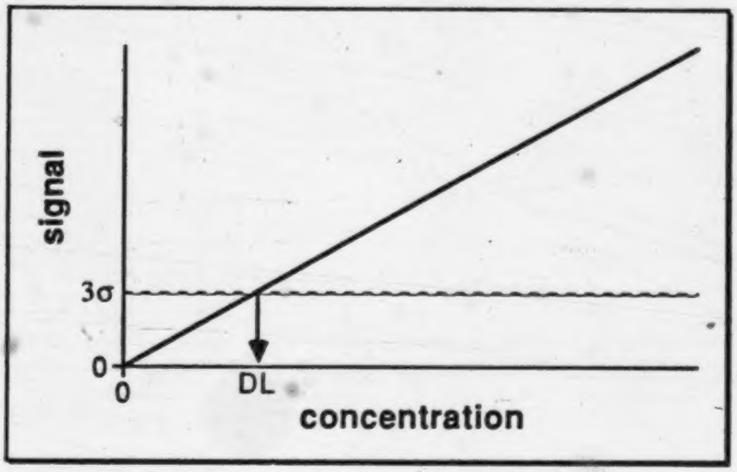


Figure 62. Graphical depiction of the detection limit definition.

ment in sensitivity can also be realized but there is no improvement in signal-to-noise ratio when flicker noise is the limiting noise.

There are two limiting cases defining the amount of improvement (dl/dl_N) possible using multiline detection limits. The best case scenario is when all the lines are equally intense and contribute equally to the combined signal. In this case, Y is equal to Nx and results in a $\sqrt{N}$ improvement in the detection limit. This is shown in Equation 8-4.

$$\frac{dl}{dl_N} = \frac{\frac{3\sigma C}{x}}{\frac{3\sqrt{N}\,\sigma C}{Y}} = \frac{Y}{\sqrt{N}\,x} = \frac{Nx}{\sqrt{N}\,x} = \sqrt{N} \qquad \text{Equation 8-4}$$

The worst case is when only one line makes a significant contribution to the total signal. Here Y is effectively x and a $\sqrt{N}$ degradation in detection limit results. This is because the noise accumulates faster than the signal.

$$\frac{dl}{dl_N} = \frac{Y}{\sqrt{N}\,x} = \frac{x}{\sqrt{N}\,x} = \frac{1}{\sqrt{N}} \qquad \text{Equation 8-5}$$

For intermediate cases the gain in detection limit is given by the general equation:

$$\frac{dl}{dl_N} = \frac{K}{\sqrt{N}} \quad \text{where } K = Y/x \qquad \text{Equation 8-6}$$

These situations were confirmed with some simple spectral simulations. These were based on 64 simulated spectral points containing 9 single point peaks with a maximum intensity of 20. A random baseline was generated with values evenly distributed between 1 and –1, the standard deviation of the baseline is 0.58.

The best case scenario is shown in Figure 63. The conventional single line detection limit calculated from the first line is 0.0864C. The multiline detection limit is 0.0290C, which results in a relative improvement of 2.98. This compares favourably with a predicted improvement of 3 (the square root of 9).

A non-optimal, but more realistic case, is shown in Figure 64 where most of the lines are much weaker than the most intense one. Again, the single line detection limit is 0.0864C while the multiline detection limit is now 0.0910C. This results in a ratio of 0.949 which represents a degradation in performance. The actual line intensities are 20, 12, 5, 4, 3, 4, 4, 3, and 3 which gives a total of 58. From this total, the expected ratio of detection limits is 0.967 which is again close to the measured value.

These simulations support our model for multiline detection limits. What remains is to apply multiline detection limits to real data.

8-2. Fe detection limit

In order to determine if there is truly any advantage to be gained from using multiline calibrations, a series of Fe ICP spectra were obtained from which both normal and multiline detection limits were calculated. The spectra were calculated from 16K interferograms acquired using the basic HeNe clock, each acquired interferogram was the result of 32 co-added scans. Solutions containing 0, 0.2, 2, and 20 mg/L Fe were used. The 15 most intense lines were used for the multiline calculations and are given in Table 17 while the single line detection limit was calculated using the 238.207 nm line.

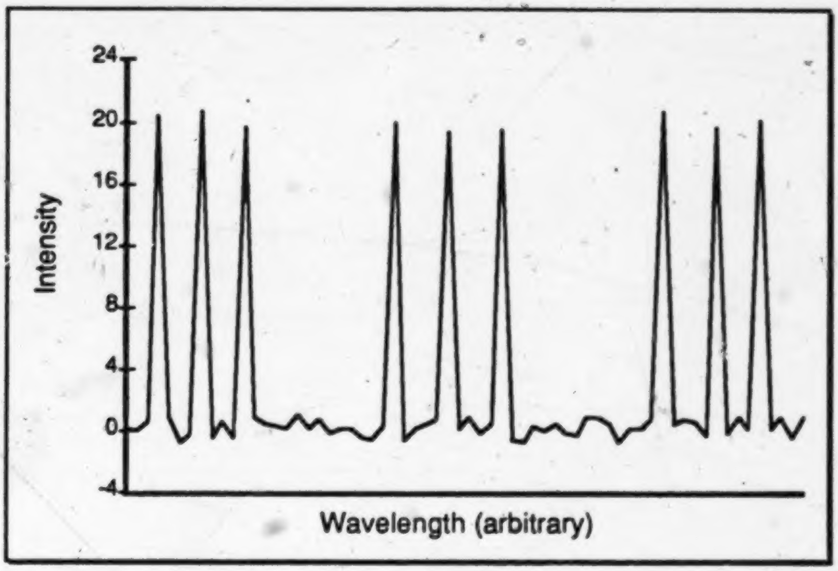


Figure 63. Simulated spectrum with equal intensity lines.

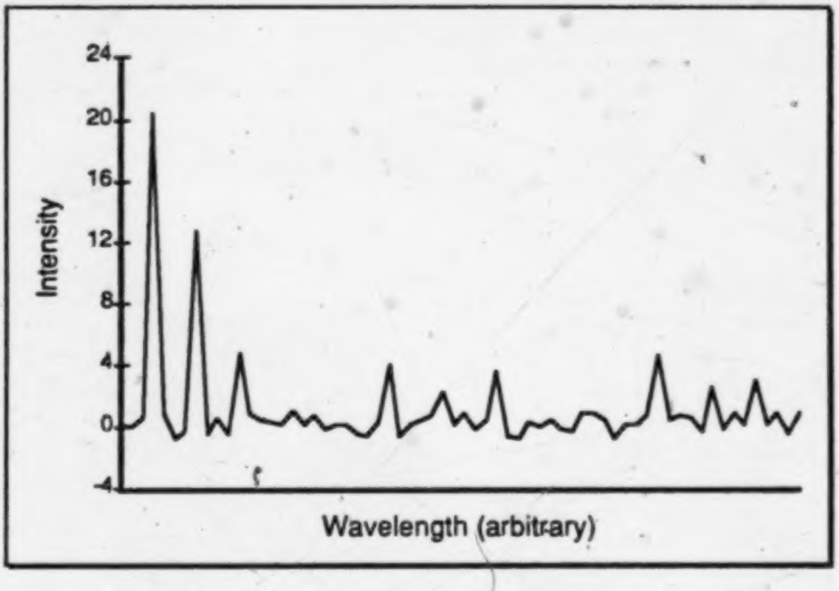


Figure 64. Simulated spectrum with only 2 major lines.

Wavelength (nm)	Relative intensity
234.349	0.325
238.207	1.000
238.863	0.191
239.562	0.683
239.924	0.198
240.488	0.329
240.666	0.150
241.052	0.219
241.107	0.117
241.331	0.099
258.588	0.276
259.837	0.302
259.940	0.816
260.709	0.232
261.187	0.357

Table 17. Fe lines used for multiline calibration.

The 15 line calibration curve is shown in Figure 65 and shows excellent linearity, the linear least squares correlation coefficient is 0.99999. The sum of the relative intensities of all the lines is 5.294, the theoretical gain is therefore $5.294/\sqrt{15}$ or 1.37. The calculated detection limits for single and multiple lines are 0.142 and 0.100 mg/L respectively which represents an actual gain of 1.42.

Closer inspection of the data in Table 17 shows that many of the lines are less than 25% of the strongest line. It is apparent that using these lines would probably contribute more noise than signal. Recalculating the multi-line detection limit using only the three most intense lines gives a new detection limit of 0.089 mg/L, a further improvement over the conventional value.

A plot of the theoretical dI/dI_N vs. N for the 15 Fe lines (Figure 66) shows the improvement in detection limit rising rapidly to a maximum at N=3 where it levels off and then slowly falls after N=8. The significance of this plot is that one cannot simply choose an arbitrarily large number of lines to be used. Instead, each element will have an optimum number of lines that can be used based on their relative intensities.

8-3. Conclusions

By itself, the use of multiple lines is not sufficient to overcome the multiplex disadvantage in our current ICP configuration. This is primarily due to the lack of sufficient numbers of strong lines in many elements. In conjunction with a quieter source, though, it may be possible to eliminate completely the multiplex disadvantage.

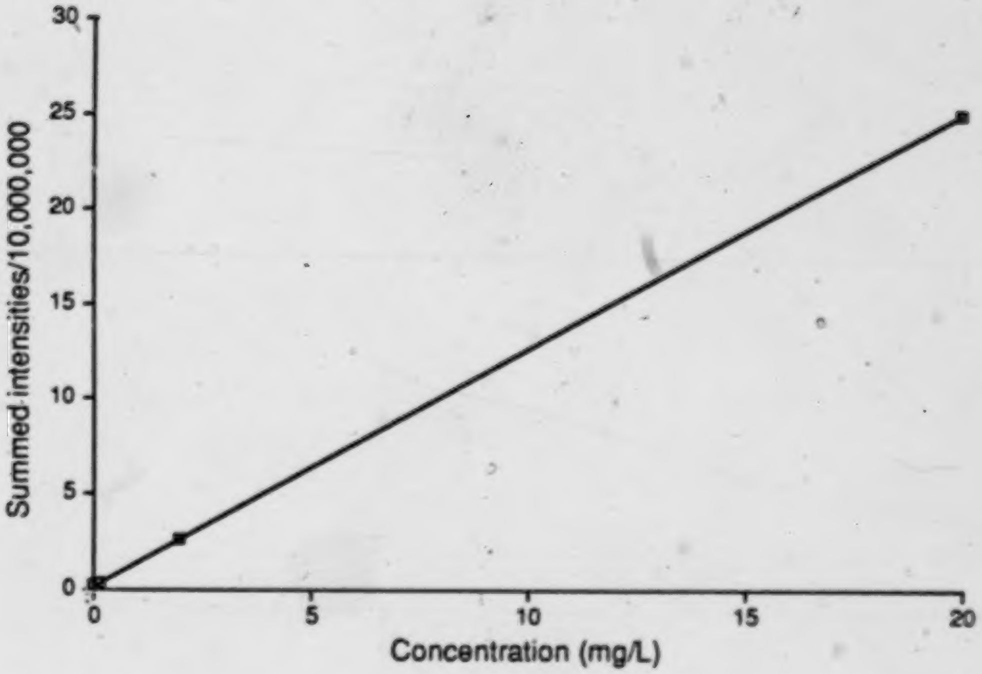


Figure 65. Multiline calibration curve for Fe.

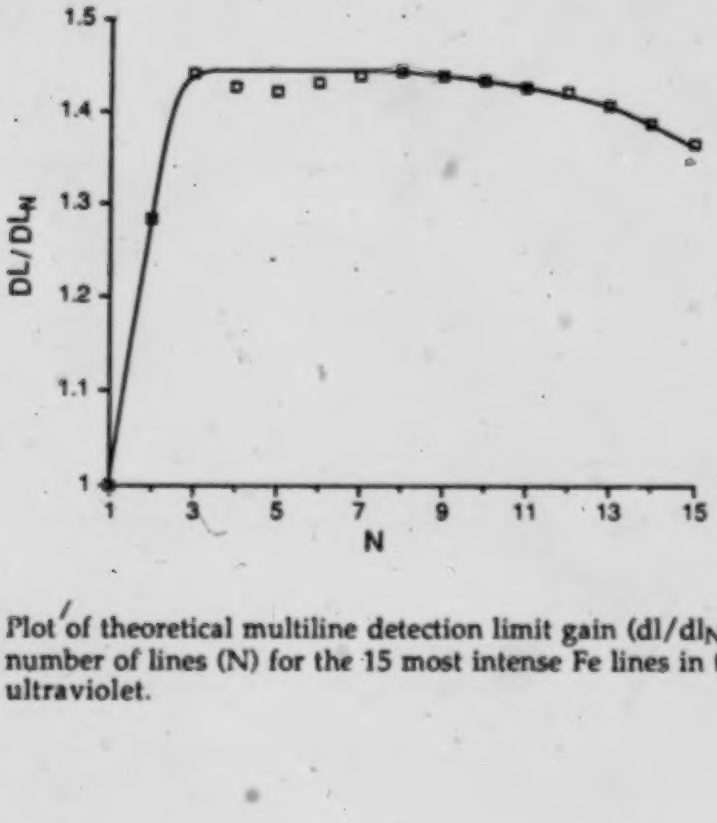


Figure 66. Plot of theoretical multiline detection limit gain (dl/dl$_N$) vs. number of lines (N) for the 15 most intense Fe lines in the ultraviolet.

It should also be noted that the use of multiple lines is not restricted to multiplex techniques such as FTS. Any technique that can measure multiple signals for a single species can use it to improve detection limits. One example is linear photodiode array systems.

Chapter 9

Summary and overall conclusions.

Solely on the basis of detection limit performance, ICP–FTS currently has a performance at least an order of magnitude worse than conventional methods. This is due to the multiplex nature of FTS which spreads noise from intense emission lines throughout the spectrum. This leads to a general increase in the background noise level which can obscure minor spectral features. In conventional dispersive systems, the noise from intense emission lines remains localized and does not interfere with neighbouring lines. This is a fundamental problem of FTS and until the noise in the ICP can be significantly reduced ICP–FTS will remain ill-favoured.

The use of multiple analytical lines can partially offset the multiplex disadvantage but the method only works for elements with several strong emission lines. Bandwidth limiting the ICP emission [18] is a more general solution to the problem of the multiplex disadvantage. The use of bandwidth limiting can improve the performance of ICP–FTS to the same level as conventional systems. The disadvantage of bandwidth limiting is the sacrifice of FTS's broad wavelength coverage.

We are currently evaluating a novel scheme for optically filtering the ICP emission while minimizing the loss of spectral coverage. The system uses two low resolution diffraction gratings and a spatial mask. The first grating is used to disperse the emission from the ICP. The spatial mask is used to select the spectral regions of interest which are then re-combined by the second grating. This "quasi-white", filtered emission is then analyzed using FTS.

The advantage of this system is that intense emission from matrix species (the major source of noise) can be eliminated using the appropriate mask while allowing the rest of the spectrum to pass through unaltered. The only degradation in performance is some loss of light through the grating system.

The major strength of ICP–FTS is its ability to rapidly acquire spectra with high wavelength accuracy. This makes it particularly well suited as a survey method. A complete 2 cm^{-1} resolution de-aliased spectrum covering the range from 158 to 316 nm is recorded in 1.625 sec. The total measurement time is determined by the number of scans that must be co-added together for an acceptable signal-to-noise level. Again, its only disadvantage is its inability to record minor spectral features because of the multiplex disadvantage.

On several occasions, the recorded spectra showed spectral lines from contaminant species even in regions that were not being investigated. When automatic peak detection software is used, the detection of contaminant species is assured. This makes ICP–FTS particularly attractive for quality control applications.

We have shown that atomic emission lines can be assigned by searching through a database of known lines, despite the possibility of aliasing. In addition, we have also demonstrated a simple expert system that can identify the species present in a spectrum on the basis of these assignments and knowledge of the prominent lines of the elements.

However, there still exists room for improvement and these will be discussed in this chapter.

9-1. Improvements in ICP performance.

In order to successfully use the ICP with FTS, the noise characteristics of the ICP must be improved in a significant manner. The two major areas for improvement are broadband noise and plasma modulation. Studies by Belchamber and Horlick [86] have revealed that the noise power spectrum of the ICP has two principle components, a continuum usually showing a strong 1/f nature and localized peak-like features representing quasi-periodic fluctuations in the emission intensity (sharp noise peaks at the power line frequency and its harmonics are also observed).

The continuum, which represents the broadband noise, was shown to have a strong dependence on sample introduction while the localized features appear to be dependent upon torch geometry and configuration. The localized noise peaks represent a modulation of the emission intensity which can be seen if the output of the PMT is directly observed on an oscilliscope. The amplitude of this modulation represents a major limiting factor on the performance of ICP–FTS.

Because the emission intensities are modulated, the interferogram becomes modulated and at low signal intensities rides on the modulation waveform. When the amplitude of the modulation is greater than that of the interferogram it dominates the signal and sets the limit to the amount of amplification that can be applied to it. This can prevent weak signals from being detected.

This modulation has been shown to be eliminated by using a tall torch [86]. This, however, can reduce the sensitivity in the uv when viewing in the conventional manner from the side. Other approaches to this problem have been the use of a laminar flow torch [105] and end on viewing of the plasma [106]. Hobbs *et al.* [89] have recently combined a tall torch with end on viewing in a study of detection limits and signal-to-noise ratios in ICP–FTS.

Improvements in sample introduction are also required to reduce the level of broadband noise. The simple needle valve nebulizer gas flow controller and natural aspiration in our system is clearly inadequate. Variations in sample uptake rate can sometimes be visually observed. This is especially true when highly concentrated sample solutions are used. Current commercial ICP spectrometers are now being delivered with mass flow controllers for the nebulizer and peristaltic pump sample delivery systems. Use of these two devices should decrease the amount of noise due to sample introduction.

Though not related to ICP performance, higher resolution analog to digital converters are also required. Current generation analytical instrumentation are using 12 and 14 bit ADC's but we feel that FTS requires 16 bits or more in order to resolve small spectral features in the presence of major emission lines and strong background emission. Fortunately, the recent large consumer demand for high speed, high quality 16 bit ADC's for digital audio applications have made them more available at lower prices and more instrumentation should begin to have them.

9-2. Software improvements.

The software for finding peaks in a spectrum must be improved, fortunately this does not appear to be an insurmountable problem. It may be possible to both distinguish unresolved peaks and ignore baseline shifts through the application of Kalman filtering techniques [107]. Taking into account variable baseline noise should also be considered. Some preliminary investigations into the applications of Kalman filters have been initiated.

The consideration of relative line ratios by the expert system should also be implemented. Because the intensity ratios are fixed, they can be used to resolve spectral overlaps which can be detected by simple line matching. The one disadvantage of using intensity ratio information is that it is specific for each particular set of ICP operating and observation conditions and is sometimes subject to chemical interferences. Unlike wavelengths, the intensities must be measured under actual ICP running conditions because literature values may not be the same.

The one major feature that is currently missing in our software is a macro command language facility. Such a capability would allow programming and the automation of repetitive tasks. It would also make remote usage possible and with some modifications execution in a multi-user shared environment.

9-3. Future directions in analytical instrumentation.

Our system is just a step towards the goal of more accessible, friendlier, more intelligent, distributed analytical instrumentation. The ultimate goal would be to integrate the software and hardware under a new, all inclusive, human-processor interface proposed by Karanassios and Horlick [54], the instrument metaphor. This metaphor forms the basis for the virtual instrument interface and an example is the LabView program from National Instruments.

Intelligent instruments are necessary in order to make them more accessible to users. This would result in increased hardware cost but the economic benefits of increased usage may offset this. Many current generation "intelligent" instruments feature embedded microprocessor controllers and are designed to operate in either a stand-alone mode or communicate with a host computer using either serial (RS-232) or parallel (IEEE-488) protocols as shown in Figure 67. The disadvantage of such a "star" configuration, is that only one user can be on the system at once. If the user is only working with a single instrument then the others remain idle, unusable by others.

If even more intelligent instruments featuring complete embedded computer systems are utilized, then it is possible to connect them as intelligent peripherals on a local area network (Figure 68). This allows more than one user on the system and better usage of the total system resources as the instruments can now shared. While only one user can be on a given instrument at a given time, it is possible to queue multiple jobs using server software. With remote sample handling the user doesn't even have to be in the same room as the instrument, the analytical chemist could conceivably perform an analysis from her/his office.

In such a distributed chemical network, each instrument would become a "chemical workstation", a peripheral device that is shared by sever-

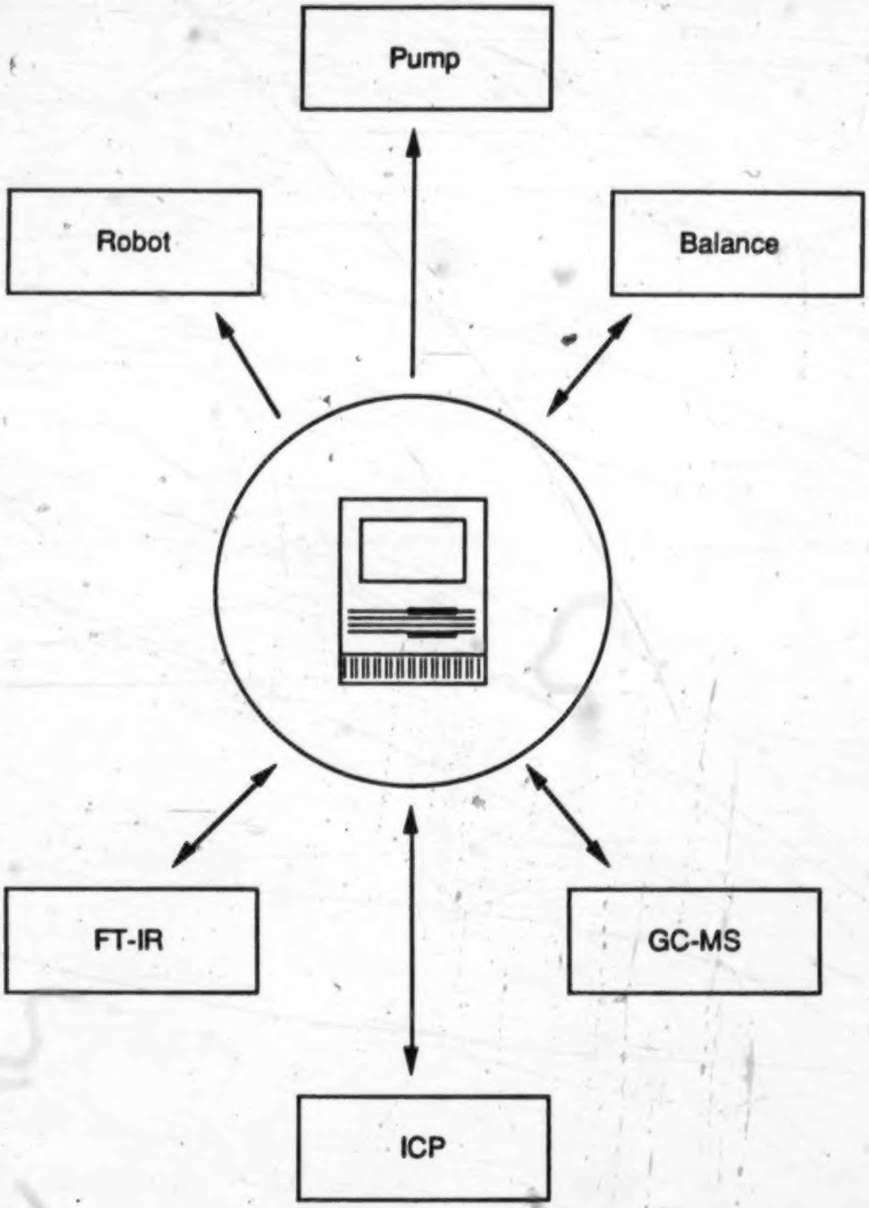

Pump
Robot
Balance
FT-IR
GC-MS
ICP

Figure 67. Conventional configuration for instrument automation.

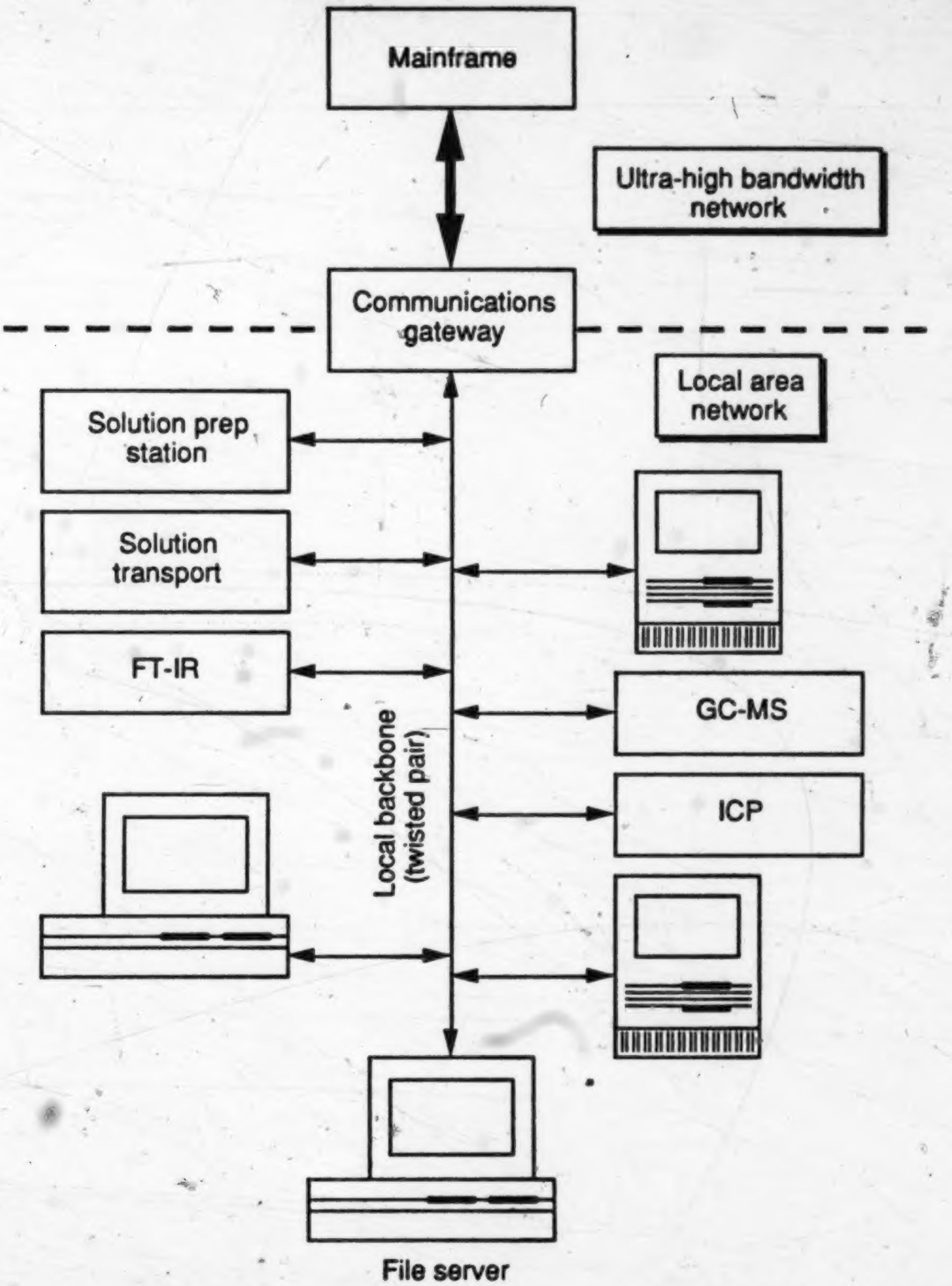

Mainframe
Ultra-high bandwidth network
Communications gateway
Local area network
Solution prep station
Solution transport
FT-IR
GC-MS
ICP
Local backbone (twisted pair)
File server

Figure 68. Distributed instrumentation with remote solution handling, "chemical workstations", and computer workstations on a local area network.

P

al analytical chemists. Such a system requires high speed local area networks such as Ethernet and powerful computer workstations running multitasking software. The major implication for hardware manufacturers is that instruments will no longer exist in isolation, they must conform to industry standards and be able to work in harmony with other manufacturers' equipment.

Instruments will become increasingly software driven, a trend already observed today. They will be entirely operated through software commands received from external computers. The major impact of software on analytical chemistry will be in the area of throughput. With an ever increasing sample load, the analytical chemist needs relief from the routine operations required by the analyses so that (s)he can concentrate on the results. This can be accomplished by automating analyses using software and computer controlled instrumentation.

Current software automates some simple repetitive tasks but more sophisticated software will be able to automate entire experiments and procedures in an intelligent manner to produce the desired results without user intervention. Intelligence is defined here as the ability to respond to unusual or unexpected conditions in a manner similar to that of a trained human. This type of automation requires the use of expert systems in a decision feedback loop controlling the analysis.

The major factor restricting the widespread use of expert systems today is legal, the question of liability must be resolved. This is the reason why current expert systems only recommend possible actions to an operator and initiate no actions by themselves. Future systems will require that the expert systems run without human supervision. This is the only way of getting maximum performance; the human ability to process information is the limiting factor.

The software would also filter and process the raw data before presenting it to the user. Intelligent software could also be used to deduce information from the data and would present that, instead, to the user. This is necessary, because the increasing number of analyses would generate a corresponding increase in data which must be interpreted. Without some sort of pre-processing, the sheer volume of data would soon overwhelm the chemist.

A more detailed block diagram of one such "intelligent" instrument unit is shown in Figure 69. The actual instrument, control software, and some of the data processing software resides in each chemical workstation. Additional data processing software and the expert system run on the user's computer. Communications between the two sections are via a local area network. A closed loop configuration where the expert system also controls the instrument is shown. The same expert system can also be controlling the solution preparation station and other instruments at the same time.

Provision should also be made for the user to be able to directly control the instruments and view the raw data using a virtual instrument interface on her/his workstation. This allows each instrument to be run manually like a "dumb" instrument. The ability to run the instruments manually adds an extra level of flexibility to the system, the user is not restricted to the capabilities of the existing software.

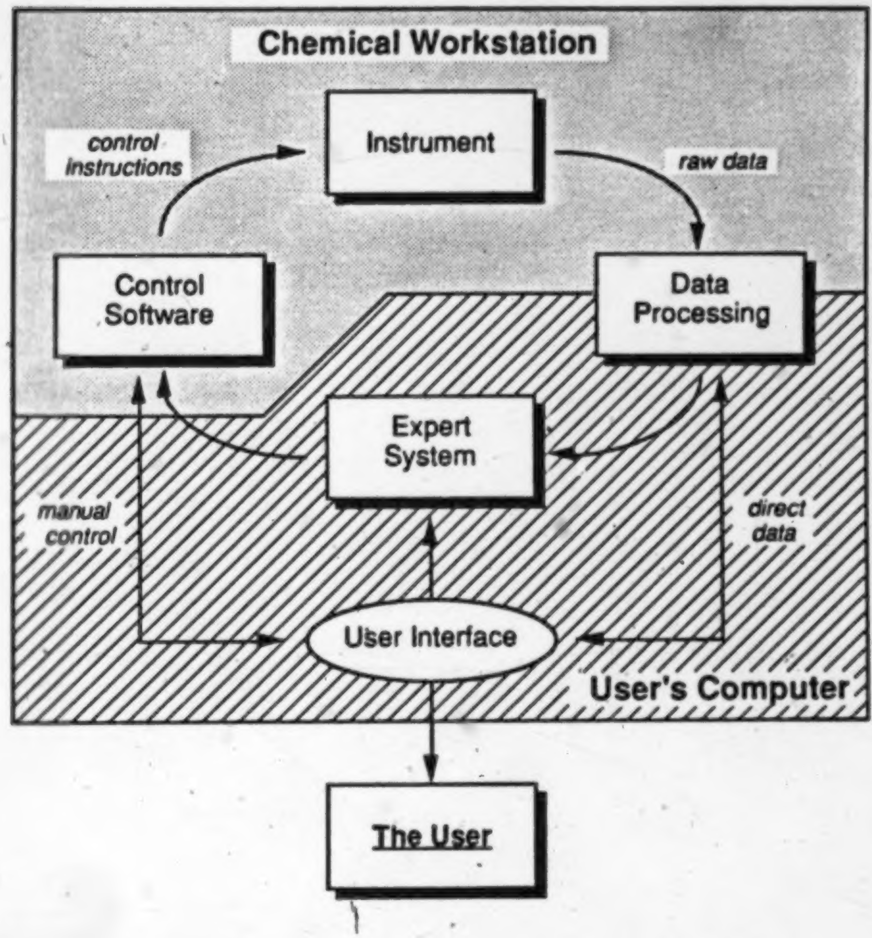

Chemical Workstation
Instrument
control instructions
raw data
Control Software
Data Processing
Expert System
manual control
direct data
User Interface
User's Computer
The User

Figure 69. Block diagram of a single intelligent instrument unit.

Prototypes of what will eventually be intelligent chemical workstations are already being developed. Williams [108] is developing microcontroller based instrumentation that use the Apple LocalTalk™ local area network for communications with the user. Software is based on an object oriented language and control is entirely through software messages.

By applying advances in both hardware and software to the design and construction of analytical instrumentation, more powerful and easier to use instruments will become available to the analytical chemist. Because the rate of technological change is accelerating, new instruments must be designed so that future developments can be incorporated into the existing system. Ultimately, the goal of a universal, totally automated, silicon (or GaAs) chemist can be achieved, hopefully before the 23rd century.

The idea is not to replace the analytical chemist but to make her/his life a little easier in the information age. By making instrumentation easier to use and intelligent enough to perform mundane routine tasks independently, the chemist is free to concentrate on the actual problem at hand. Instrumentation would then become more than a tool, it would be the chemist's assistant.

Bibliography

1. J. Brault, J. Opt. Soc. Amer., 66, 1081(1976).

2. P. Luc and S. Gerstenkorn, Appl. Opt., 17, 1327(1978).

3. M. Parsons and B. Palmer, Spectrochim. Acta, 43B, 75(1988).

4. A. Thorne, C. Harris, I. Wynne-Jones, R. Learner, and G. Cox, J. Phys. E., 20, 54(1987).

5. G. Horlick, Appl. Spectrosc., 22, 617(1968).

6. G. Horlick and H, Malmstadt, Anal. Chem., 42, 1361(1970).

7. G. Horlick, Anal. Chem., 43, 61A(1971).

8. G. Horlick, Anal. Chem, 44, 943(1972).

9. E. Codding and G. Horlick, Appl. Spectrosc., 27, 85(1973).

10. G. Horlick and W. Yuen, Anal. Chem., 47, 775A(1975).

11. W. Yuen and G. Horlick, Anal. Chem., 49, 1446(1977).

12. G. Horlick and W. Yuen, Appl. Spectrosc., 32, 38(1978).

13. G. Horlick, W. Yuen, and R. Hall, "Atomic emission spectrochemical measurements with a Fourier transform spectrometer", "Fourier transform infrared spectroscopy", vol. 3, J. Ferraro, L. Basile, and A. Mantz, eds., Academic Press, New York (1982).

14. R. Ng and G. Horlick, Spectrochim. Acta, 36B, 529(1981).

15. E. Stubley and G. Horlick, Appl. Spectrosc., 38, 162(1984).

16. E. Stubley and G. Horlick, Appl. Spectrosc., 39, 800(1985).

17. E. Stubley and G. Horlick, Appl. Spectrosc., 39, 805(1985).

18. E. Stubley and G. Horlick, Appl. Spectrosc., 39, 811(1985).

19. R. Ng and G. Horlick, Appl. Spectrosc., 39, 834(1985).

20. R. Ng and G. Horlick, Appl. Spectrosc., 39, 841(1985).

21. S. Marra and G. Horlick, Appl. Spectrosc., 40, 804(1986).

22. G. Horlick, B. Todd, G. King, Paper #001 presented at the 1988 Pittsburgh Conference, Feb. 22 – 26, 1988.

23. P. Griffiths, Appl. Spectrosc., 29, 11(1975).

24. P. Zarchon, MacTutor, Sept. 1987, pg. 78.

25. Los Alamos FTS Newsletter, July 1988.

26. "Inside Macintosh", vols. 1 – 5, Addison-Wesley, Reading MA.

27. S. Chernicoff, "Macintosh Revealed", vols. 1 and 2, Hayden, Hasbrouck Heights NJ, 1986.

28. S. Knaster, "How to Write Macintosh Software", Hayden, Hasbrouck Heights NJ, 1986.

29. S. Knaster, "Macintosh Programming Secrets", Addison-Wesley, Reading MA, 1988.

30. J. Takatsuka, F. Huxman, and D. Burnard, "Using the Macintosh Toolbox with C", Sybex, Berkeley CA, 1986.

31. K. Mathews and J. Friedland, "Encyclopedia Mac ROM", Brady, New York NY, 1988.

32. D. Weston, "The Complete Book of Macintosh Assembly Language Programming", vols. 1 and 2, Scott, Foresman and Company, Glenview IL, 1986.

33. "Apple Numerics Manual", Addison-Wesley, Reading MA, 1986.

34. "Designing Cards and Drivers for Macintosh II and Macintosh SE", Addison-Wesley, Reading MA, 1987.

35. A. Michelson, Phil. Mag. (5), **34**, 280(1892).

36. Lord Rayleigh, Phil. Mag. (5), **34**, 407(1892).

37. H. Lewis and C. Papadimitriou, "The Efficiency of Algorithms", Scientific American, Jan. 1978, pg. 96.

38. J. Cooley and J. Tukey, Math. Comput., **19**, 297(1965).

39. G. Bergland, Math. Comput., **22**, 275(1968).

40. H. Nussbaumer, "Fast Fourier Transform and Convolution Algorithms", Springer-Verlag, Berlin, 1982, pp 125-133.

41. S. Winograd, Math. Comput., **32**, 175(1978).

42. H. Silverman, IEEE Trans. Acoust., Speech, Signal Processng, **ASSP-25**, 152(1977). Corrections in IEEE Trans. Acoust., Speech, Signal Processing, **ASSP-26**, 268(1978).

43. S. Zohar, IEEE Trans. Acoust., Speech, Signal Processing, **ASSP-27**, 409(1979).

44. L. Morris, IEEE Trans. Acoust., Speech, Signal Processing, **ASSP-26**, 141(1978).

45. R. Hartley, Proc. IRE, **30**, 144(1942).

46. R. Bracewell, J. Opt. Soc. Am., **73**, 1832(1983).

47. R. Bracewell, Proc. IEEE, **72**, 1010(1984).

48. M. O'Neill, "Faster Than Fast Fourier", Byte, April 1988, pg. 293.

49. "Motorola MC68881 Floating Point Coprocessor User's Manual", Motorola, 1985.

50. R. Eckhouse and L. Morris, "Minicomputer Systems", Prentice-Hall, Inc, Englewood Cliffs, 1979, pg. 289.

51. G. Fox and P. Messina, "Advanced Computer Architectures", Scientific American, Oct. 1987, pg. 67.

52. J. D. Foley, "Interfaces for Advanced Computing", Scientific American, Oct. 1987, pg. 127.

53. A. Kay, "Computer Software", Scientific American, Sept. 1984, pg. 53.

54. V. Karanassios and G. Horlick, Appl. Spectrosc., **41**, 360(1987).

55. W. Barnett, Anal. Chem., **60**, 1169A(1988).

56. S. Brand, "The Media Lab", Viking Penguin Inc., New York, 1987, pp. 45-50.

57. R. Wolff, "The Visualization Challenge in the Physical Sciences", Computers in Science, Jan/Feb 1988, pg. 16.

58. S. Brand, "The Media Lab", Viking Penguin Inc., New York, 1987, pp. 110-116.

59. S. Brand, "The Media Lab", Viking Penguin Inc., New York, 1987, pp. 138-140.

60. J. Carroll, "Presentation and Form in User-Interface Architecture", Byte, Dec. 1983, pg. 113.

61. D. Smith, C. Irby, R. Kimball, B. Verplank, and E. Harslem, "Designing the Star User Interface", Byte, April 1982, pg. 242.

62. G. Sobelman and D. Krekelberg, "Advanced C: Techniques and Applications", Que Corp., Indianapolis, 1985, pp. 145-157.

63. Apple Computer, Inc., "Human Interface Guidelines: The Apple Desktop Interface", Addison-Wesley Publishing Company, Inc., Reading, 1987, pp. 20-38.

64. "Imitation or Infringement?", Time, April 4, 1988, pg. 53.

65. M. Bogdan, Am. Lab. **18**(10), 69(1986).

66. J. Levkov, J. Chem. Ed., **64**, 31(1987).

67. M. Salit, Anal. Chem., 60, 731A(1988).

68. T. O'Haver, Am. Lab. 18(2), 22(1986).

69. C. Gerson and R. Love, Anal. Chem., 59, 1031A(1987).

70. Apple Computer, Inc., "Human Interface Guidelines: The Apple Desktop Interface", Addison-Wesley Publishing Company, Inc., Reading, 1987, pp. 1-17.

71. D. Goodman, "The Two Faces of HyperCard", MacWorld, Oct. 1987, pg. 125.

72. S. Borman, Anal. Chem., 57, 983A(1985).

73. K. Schmucker, "Object-Oriented Programming for the Macintosh", Hayden Book Company, Hasbrouck Heights, 1986.

74. M. Olan, Computer Language, May 1988, pg. 36.

75. M. Wang and A. Marshall, Anal. Chem., 60, 341(1988).

76. R. Hawkins, M. Hoke, and J. Shaw, Appl. Spectrosc., 37, 134(1983).

77. "CRC Handbook of Laser Science and Technology, Vol. 2: Gas Lasers", M. Weber (ed.), CRC Press, Boca Raton, 1982, pg. 59.

78. "CRC Handbook of Chemistry and Physics, 57th ed.", R. Weast (ed.), CRC Press, Boca Raton, Boca Raton, 1976, pg. E-209.

79. K. Lepla and G. Horlick, paper H1.1 presented at CSI XXV in Toronto, July 21-26, 1987.

80. P. Boumans and J. Vrakking, Spectrochim. Acta, **40B**, 1085(1985).

81. H. Kawaguchi, Y. Oshio, and A. Mizuike, Spectrochim. Acta, **37B**, 809(1982).

82. P. Boumans and J. Vrakking, Spectrochim. Acta, **39B**, 1239(1984).

83. J. McLaren and J. Mermet, Spectrochim. Acta, **39B**, 1307(1984).

84. H. Malmstadt, C. Enke, S. Crouch, and G. Horlick, "Electronic Measurement for Scientists", W. A. Benjamin, Menlo Park, 1974.

85. R. Sing and J. Hubert, J. Anal. At. Spectrosc., **3**, 835(1988).

86. R. Belchamber and G. Horlick, Spectrochim. Acta, **37B**, 17(1982).

87. J. Winefordner, et al, Spectrochim. Acta, **31B**, 1(1976).

88. E. Voigtman and J. Winefordner, Appl. Spectrosc., **41**, 1182(1987).

89. P. Hobbs, D. Spillane, R. Snook, A. Thorne, J. Anal. At. Spectrosc., **3**, 543(1988).

90. E. Voigtman and J. Winefordner, Anal. Chem., **59**, 1364(1987).

91. G. Long and J. Winefordner, Anal. Chem., **55**, 712A(1983).

92. R. Winge, V. Fassel, and M. Edelson, Spectrochim. Acta, **43B**, 85(1988).

93. P. Boumans, Spectrochim. Acta, **43B**, 5(1988).

94. M. Parsons, A. Forster, and D. Anderson, "An Atlas of Spectral Interferences in ICP Spectroscopy", Plenum, New York, 1980.

95. R. Winge, V. Fassel, V. Peterson, and M. Floyd, "Inductively Coupled Plasma-Atomic Emission Spectroscopy. An Atlas of Spectral Information.", Elsevier, Amsterdam, 1985.

96. A. Zaidel, V. Prokofjev, S. Raiskii, V. Slavnyi, and E. Shreider, "Tables of Spectral Lines", IFI/Plenum, New York, 1970.

97. M. Blades, Spectrochim. Acta, **43B**, 35(1988).

98. L. Burton and M. Blades, Spectrochim. Acta, **41B**, 1063(1986).

99. C. Trassy and J. Robin, Spectrochim. Acta, **43B**, 79(1988).

100. L. Fernando, Anal. Chem., **56**, 1970(1984).

101. M.-A. Vaughan and G. Horlick, J. Anal. At. Spectrosc., in press.

102. V. Karanassios and G. Horlick, Appl. Spectrosc., **40**, 813(1986).

103. D. Spillane, R. Snook, A. Thorne, and J. Wheaton, J. Anal. At. Spectrom., **2**, 591(1987).

104. L. Layman, paper K76 presented at FACSS XV in Boston, Oct. 30-Nov. 4, 1988.

105. J. Davies and R. Snook, J. Anal. At. Spectrom., **2**, 27(1987).

106. L. Faires, T. Bieniewski, C. Apel, and T. Niemczyk, Appl. Spectrosc., **39**, 5(1985).

107. S. Bialkowski, Anal. Chem., **60**, 403A(1988).

108. R. Williams, paper K53 presented at FACSS XV in Boston, Oct. 30-Nov. 4, 1988.

Appendix A

National Instruments MIO-16 initialization program.

File: InitMIO.pas

```
PROGRAM InitMIO;

{  TML Pascal source code for an INIT that initializes a National
   Instruments MIO-16 board upon start-up.

   Copyright 1988 - Greg King

   Version 1.10, Sept 11, 1988                                   }


{$H 'PAS$PROCHeader'}        {Use custom header}
{$C 'INIT' 10 'InitMIO'}     {Resource type, flags, and name}
{$T 'INIT' 'iMIO'}           {File type and creator}
{$B+}                        {Set bundle bit}
{$L InitMIO.rsrc }           {Copy additional resources from}

USES MacIntf,Library1,National;
{$U ShowINIT}                {Seperate library routine}

{ ShowINIT will draw the icon in the lower left
  hand side of the screen during boot.                      }
PROCEDURE ShowINIT(iconID,moveX: INTEGER); EXTERNAL;

{ The PAS$PROCHeader uses PROC as its entry point. }
PROCEDURE PROC;
VAR
   myWorld: SysEnvRec;
   slot: INTEGER;
   err: OSErr;
BEGIN
{ The resource is not detatched since there's no need to
  keep it around after it's been executed. }
err:=SysEnvirons(1,myWorld);
IF (err=noErr) AND (myWorld.machineType=envMacII) THEN
   BEGIN
   FOR slot:=1 TO 6 DO           {Loop through slots}
      IF CheckSlot(slot) THEN    {Initialize if slot has ADC}
         InitMIO16(slot);
   ShowINIT(128,-1);
   END;
END; {of PROC}

{ No main procedure allowed. }
BEGIN
END.
```

Appendix B

National Instruments MIO-16 interface code.

File: National.pas

```
UNIT National;

{ Interface routines for National Instruments MIO-16 ADC card }

INTERFACE

USES MacIntf,Library1;

{$U NationalGlue}             {Low level assembly glue}

FUNCTION  IsADC(sBlock: spBlockPtr): BOOLEAN;
FUNCTION  CheckSlot(i: INTEGER): BOOLEAN;

PROCEDURE InitMIO16(slot: INTEGER);
PROCEDURE Init9513(slot: INTEGER);
PROCEDURE InitCtr(slot,ctr: INTEGER);
FUNCTION  DAQ(slot,channel,gain,time,interval: INTEGER; count: LONGINT;
              xtrigger,xgate,xclock,xsense,xlevel,bipolar,avg: BOOLEAN;
              p: Ptr): OSErr;
FUNCTION  QuickDAQ(slot,channel,gain,time,interval: INTEGER; count: LONGINT;
                   xtrigger,xgate,xclock,xsense,xlevel,bipolar,avg: BOOLEAN;
                   p: Ptr): OSErr;
FUNCTION  FTDAQ (slot,channel,gain: INTEGER; count: LONGINT;
                avg,bFwd: BOOLEAN; p: Ptr): OSErr;

IMPLEMENTATION

{High level routines}

FUNCTION IsADC ((sBlock: spBlockPtr): BOOLEAN) ;
VAR
   myBlock: spBlock;
   name: Str255;
   err: OSErr;
BEGIN
IsADC:=FALSE;
myBlock:=sBlock^;
WITH myBlock DO
   BEGIN
   {Get board name from Board resource}
   spID:=2;
   err:=sGetCString(@myBlock);
   IF err=noErr THEN
      BEGIN
      CToPStr(Ptr(spResult),name); {convert to Pascal string}
      IF POS('NB-MIO',name)<>0 THEN
         IsADC:=TRUE
      END;
   END;
END; {of IsADC}
```

```
FUNCTION CheckSlot {(i: INTEGER): BOOLEAN} ;
VAR
   myBlock: spBlock;
   err: OSErr;
BEGIN
CheckSlot:=FALSE;
WITH myBlock DO
   BEGIN
   {Get Board resource}
   spSlot:=i+8;
   spID:=0;
   spCategory:=1;
   spCType:=0;
   spDrvrSW:=0;
   spDrvrHW:=0;
   spHWDev:=0;
   spTBMask:=0;
   err:=sNextTypesRsrc(@myBlock);
   {Is there a board resource for the selected slot}
   IF (err=noErr) AND (spSlot=(i+8)) THEN
      IF IsADC(@myBlock) THEN
         CheckSlot:=TRUE;
   END;
END; {of CheckSlot}

{Medium level routines}

{Routine to initialize all card functions}
PROCEDURE InitMIO16 {(slot: INTEGER)} ;
VAR
   slotBase: LONGINT;
   comReg1,comReg2,muxGain,ADclear: ^INTEGER;
BEGIN
slotBase:=slot*$100000+$800000;
comReg1:=POINTER(slotBase);
comReg2:=POINTER(slotBase+4);
muxGain:=POINTER(slotBase+$C);
ADclear:=POINTER(slotBase+$18);
comReg1^:=0;
comReg2^:=0;
muxGain^:=0;
Init9513(slot);
ADclear^:=0;
END; {of InitMIO16}

{Routine to initialize a single counter}
PROCEDURE InitCtr {(slot,ctr: INTEGER)} ;
VAR
   slotBase: LONGINT;
   am9513dat,am9513com: ^INTEGER;
BEGIN
slotBase:=slot*$100000+$800000;
am9513dat:=POINTER(slotBase+$30);
am9513com:=POINTER(slotBase+$34);
am9513com^:=$FF00+ctr;
am9513dat^:=4;
am9513com^:=$FF08+ctr;
am9513dat^:=3;
END; {of InitCtr}

{Routine to initialize all counters}
PROCEDURE Init9513 {(slot: INTEGER)} ;
VAR
```

```
   slotBase: LONGINT;
   am9513dat,am9513com: ^INTEGER;
   i: INTEGER;
BEGIN
slotBase:=slot*$100000+$800000;
am9513dat:=POINTER(slotBase+$30);
am9513com:=POINTER(slotBase+$34);
am9513com^:=$FFFF;
am9513com^:=$FFEF;
am9513com^:=$FF17;
am9513dat^:=$F000;
FOR i:=1 TO 5 DO
   InitCtr(slot,i);
am9513com^:=$FF5F;
END; {of Init9513}

{ An external assembly language ADC service is needed for data
  acquisition to run at full speed (100 kHz).  Pascal code
  produced buffer overflows at 38-50 usec.

  The assembly source is in NationalGlue.Asm  }

FUNCTION ServiceDAQ(slot: INTEGER; count: LONGINT p: Ptr): OSErr;
   EXTERNAL;

FUNCTION AddDAQ(slot: INTEGER; count: LONGINT p: Ptr): OSErr;
   EXTERNAL;

FUNCTION AddDAQ2(slot: INTEGER; count: LONGINT p: Ptr): OSErr;
   EXTERNAL;

FUNCTION  DAQ ((slot,channel,gain,time,interval: INTEGER; count: LONGINT;
               xtrigger,xgate,xclock,xsense,xlevel,bipolar,avg: BOOLEAN;
               p: Ptr): OSErr) ;
VAR
   slotBase: LONGINT;
   comReg1,muxGain,ADclear,Din,Strobe: ^INTEGER;
   am9513dat,am9513com: ^INTEGER;
   StartDAQ: ^INTEGER;
   theStatus,i: INTEGER;
BEGIN
slotBase:=slot*$100000+$800000;
comReg1:=POINTER(slotBase);
muxGain:=POINTER(slotBase+$C);
ADclear:=POINTER(slotBase+$18);
am9513dat:=POINTER(slotBase+$30);
am9513com:=POINTER(slotBase+$34);
StartDAQ:=POINTER(slotBase+$14);
Din:=POINTER(slotBase+$38);
Strobe:=POINTER(slotBase+$1C);

muxGain^:=gain*64+channel; {set ADC gain and channel}
IF NOT xclock THEN
   BEGIN
   CASE time OF
      0: i:=$8B25;
      1: i:=$8C25;
      2: i:=$8D25;
      3: i:=$8E25;
      4: i:=$8F25;
      5: i:=$8525;
      END; {CASE time}
   IF xgate THEN
      i:=i-$1000;
   am9513com^:=$FF03;
   am9513dat^:=i;
   am9513com^:=$FF0B;
   am9513dat^:=2;
   am9513com^:=$FF44;
   am9513com^:=$FFF3;
   am9513dat^:=interval;
   am9513com^:=$FF24;
   END;
IF (NOT xgate) AND (NOT xclock) THEN
   BEGIN
   am9513com^:=$FF04;
   am9513dat^:=$1001;
   am9513com^:=$FF0C;
   am9513dat^:=LOWORD(count);
   IF count<=65536 THEN
      am9513com^:=$FF68
   ELSE
      BEGIN
      am9513com^:=$FF48;
      am9513dat^:=0;
      am9513com^:=$FF28;
      am9513com^:=$FF05;
      am9513dat^:=1;
      am9513com^:=$FF0D;
      am9513dat^:=HIWORD(count)+1;
      am9513com^:=$FF70;
      END;
   END;
i:=0;
IF NOT xclock THEN
   i:=i+$10;
IF NOT bipolar THEN
   i:=i+1;
IF (count>65536) AND (NOT xgate) AND (NOT xclock) THEN {set 32 bit count mode}
   i:=i+2;
comReg1^:=i;
IF xsense THEN {Use DIO A0 to gate acquisition}
   BEGIN
   {Pressing cmd-period will cancel}
   REPEAT
      i:=BitAnd(Din^,1);
      IF i=ORD(xlevel) THEN
         BEGIN
         IF CmdPeriod THEN
            BEGIN
            DAQ:=2;
            InitMIO16(slot);
            EXIT(DAQ);
            END;
         END
      ELSE
         BEGIN
         {Read line twice for noise immunity}
         i:=BitOr(i,BitAnd(Din^,1));
         END;
   UNTIL i=ORD(NOT xlevel);
   REPEAT
      i:=BitAnd(Din^,1);
      IF i=ORD(NOT xlevel) THEN
         BEGIN
         IF CmdPeriod THEN
            BEGIN
            DAQ:=3;
            InitMIO16(slot);
            EXIT(DAQ);
            END;
         END
      ELSE
         BEGIN
         i:=BitAnd(i,BitAnd(Din^,1));
         END;
   UNTIL i=ORD(xlevel);
   END;
ADclear^:=0;
Strobe^:=0;
IF (NOT xtrigger) AND (NOT xgate) AND (NOT xclock) THEN
   StartDAQ^:=0;

IF avg THEN
   DAQ:=AddDAQ(slot,count,p)
ELSE
   DAQ:=ServiceDAQ(slot,count,p);

{ Reset the counters after data acquisition. }
IF NOT xclock THEN
   BEGIN {reset counter 3}
   am9513com^:=$FFC4;
   am9513com^:=$FF03;
   am9513dat^:=4;
   am9513com^:=$FF0B;
   am9513dat^:=3;
   am9513com^:=$FF44;
   am9513com^:=$FF44;
   END;
IF (NOT xgate) AND (NOT xclock) THEN
   BEGIN {reset counter 4}
   am9513com^:=$FFC8;
   am9513com^:=$FF04;
   am9513dat^:=4;
   am9513com^:=$FF0C;
   am9513dat^:=3;
   am9513com^:=$FF48;
   am9513com^:=$FF48;
   IF count>65536 THEN
      BEGIN {reset counter 5}
      am9513com^:=$FFD0;
      am9513com^:=$FF05;
      am9513dat^:=4;
      am9513com^:=$FF0D;
      am9513dat^:=3;
      am9513com^:=$FF50;
      am9513com^:=$FF50;
      END;
   END;
ADclear^:=0;
END; {of DAQ}

{Use this routine if you need faster response when gating acquisition
 through DIO line A0.}
FUNCTION  QuickDAQ ((slot,channel,gain,time,interval: INTEGER; count: LONGINT;
                   xtrigger,xgate,xclock,xsense,xlevel,bipolar,avg: BOOLEAN;
                   p: Ptr): OSErr) ;
VAR
   slotBase: LONGINT;
   comReg1,muxGain,ADclear,Din,Strobe: ^INTEGER;
   am9513dat,am9513com: ^INTEGER;
   StartDAQ: ^INTEGER;
   theStatus,i: INTEGER;
BEGIN
slotBase:=slot*$100000+$800000;
comReg1:=POINTER(slotBase);
muxGain:=POINTER(slotBase+$C);
ADclear:=POINTER(slotBase+$18);
am9513dat:=POINTER(slotBase+$30);
am9513com:=POINTER(slotBase+$34);
StartDAQ:=POINTER(slotBase+$14);
Din:=POINTER(slotBase+$38);
Strobe:=POINTER(slotBase+$1C);

muxGain^:=gain*64+channel; {set ADC gain and channel}
IF NOT xclock THEN
   BEGIN
   CASE time OF
      0: i:=$8B25;
      1: i:=$8C25;
      2: i:=$8D25;
      3: i:=$8E25;
      4: i:=$8F25;
      5: i:=$8525;
      END; {CASE time}
   IF xgate THEN
      i:=i-$1000;
   am9513com^:=$FF03;
   am9513dat^:=i;
   am9513com^:=$FF0B;
   am9513dat^:=2;
   am9513com^:=$FF44;
   am9513com^:=$FFF3;
   am9513dat^:=interval;
   am9513com^:=$FF24;
   END;
IF (NOT xgate) AND (NOT xclock) THEN
   BEGIN
   am9513com^:=$FF04;
   am9513dat^:=$1001;
   am9513com^:=$FF0C;
   am9513dat^:=LOWORD(count);
   IF count<=65536 THEN
      am9513com^:=$FF68
   ELSE
      BEGIN
      am9513com^:=$FF48;
      am9513dat^:=0;
      am9513com^:=$FF28;
      am9513com^:=$FF05;
      am9513dat^:=1;
      am9513com^:=$FF0D;
      am9513dat^:=HIWORD(count)+1;
      am9513com^:=$FF70;
      END;
   END;
i:=0;
IF NOT xclock THEN
   i:=i+$10;
IF NOT bipolar THEN
   i:=i+1;
IF (count>65536) AND (NOT xgate) AND (NOT xclock) THEN {set 32 bit count mode}
   i:=i+2;
comReg1^:=i;
IF xsense THEN
   BEGIN
   REPEAT
      i:=BitAnd(Din^,1);
      IF i=ORD(xlevel) THEN
         BEGIN
         IF CmdPeriod THEN
            BEGIN
            QuickDAQ:=2;
            InitMIO16(slot);
            EXIT(QuickDAQ);
            END;
         END
      ELSE
         BEGIN
         i:=BitOr(i,BitAnd(Din^,1));
         END;
   UNTIL i=ORD(NOT xlevel);
   {Can't cancel once we reach this point, the CmdPeriod
    routine eats up too many clock cycles.}
   REPEAT
      i:=BitAnd(Din^,1);
      IF i=ORD(NOT xlevel) THEN
         BEGIN
         END
      ELSE
         BEGIN
         i:=BitAnd(i,BitAnd(Din^,1));
         END;
   UNTIL i=ORD(xlevel);
   END;
ADclear^:=0;
Strobe^:=0;
IF (NOT xtrigger) AND (NOT xgate) AND (NOT xclock) THEN
   StartDAQ^:=0;

IF avg THEN
   QuickDAQ:=AddDAQ(slot,count,p)
ELSE
   QuickDAQ:=ServiceDAQ(slot,count,p);

{ Reset the counters after data acquisition. }
IF NOT xclock THEN
   BEGIN {reset counter 3}
   am9513com^:=$FFC4;
   am9513com^:=$FF03;
   am9513dat^:=4;
   am9513com^:=$FF0B;
   am9513dat^:=3;
   am9513com^:=$FF44;
   am9513com^:=$FF44;
   END;
IF (NOT xgate) AND (NOT xclock) THEN
   BEGIN {reset counter 4}
   am9513com^:=$FFC8;
   am9513com^:=$FF04;
   am9513dat^:=4;
   am9513com^:=$FF0C;
   am9513dat^:=3;
   am9513com^:=$FF48;
   am9513com^:=$FF48;
   IF count>65536 THEN
      BEGIN {reset counter 5}
      am9513com^:=$FFD0;
      am9513com^:=$FF05;
      am9513dat^:=4;
```

```
            am9513com^:=$FF0D;
            am9513dat^:=3;
            am9513com^:=$FF50;
            am9513com^:=$FF50;
            END;
        END;
    ADclear^:=0;
    END; {of QuickDAQ}

    {This routine has been optimized for our instrument.}
    FUNCTION  FTDAQ ((slot,channel,gain: INTEGER; count: LONGINT;
                    avg,bFwd: BOOLEAN; p: Ptr): OSErr) ;
    VAR
        anEvent: EventRecord;
        slotBase: LONGINT;
        comReg1,muxGain,ADclear,Din,Strobe: ^INTEGER;
        theStatus,i: INTEGER;

        FUNCTION Wait(address: Ptr; mask: INTEGER): INTEGER;
            INLINE
                $301F, {MOVE.W (A7)+,D0}
                $205F, {MOVEA.L (A7)+,A0}
                $C050, {AND.W (A0),D0}
                $C050,
                $3E80; {MOVE.W D0,(A7)}

    BEGIN
    slotBase:=slot*$100000+$800000;
    comReg1:=POINTER(slotBase);
    muxGain:=POINTER(slotBase+$C);
    ADclear:=POINTER(slotBase+$18);
    Din:=POINTER(slotBase+$38);
    Strobe:=POINTER(slotBase+$1C);

    muxGain^:=gain*64+channel; {set ADC gain and channel}
    comReg1^:=$10;

    IF NOT bFwd THEN
        p:=Ptr(ORD4(p)+4*count);

    REPEAT
        i:=BitAnd(Din^,1);
        IF i=1 THEN
            BEGIN
            IF EventAvail(keyDownMask,anEvent) THEN
                IF CmdPeriod THEN
                    BEGIN
                    FTDAQ:=2;
                    InitMIO16(slot);
                    EXIT(FTDAQ);
                    END;
            END
        ELSE
            BEGIN
            i:=BitOr(i,BitAnd(Din^,1));
            END;
    UNTIL i=0;

    REPEAT
    UNTIL Wait(Ptr(Din),1)=1;

    ADclear^:=0;
    Strobe^:=0;
```

```
IF count>0 THEN
   IF avg THEN
      IF bFwd THEN
         FTDAQ:=AddDAQ(slot,count,p)
      ELSE
         FTDAQ:=AddDAQ2(slot,count,p)
   ELSE
      FTDAQ:=ServiceDAQ(slot,count,p)
ELSE
   FTDAQ:=0;

ADclear^:=0;
END; {of FTDAQ}

END.
```

File: NationalGlue.asm

```
;
;  Assembly language routines to service data acquisition
;  from National Instruments ADC board.
;
;  Greg King
;  January 28, 1988
;
;  modified June 27/88 - AddDAQ2 moves backwards through array
;

; Stackframe equates

func            EQU     18                  ; function result, OSErr
slot            EQU     16                  ; INTEGER
count           EQU     12                  ; LONGINT
p               EQU     8                   ; Ptr
return          EQU     4                   ; return address
A6Link          EQU     0                   ; previous stack frame

locals          EQU     0                   ; size of locals

, MIO-16 equates

statReg         EQU     0                   ; offset of MIO-16 status register
FIFO            EQU     $2C                 ; offset of ADC buffer
cAvail          EQU     13                  ; bit position of Conv Avail flag
error           EQU     $300                ; mask for overrun and overflow flags

;
;  FUNCTION ServiceDAQ(slot: INTEGER; count: LONGINT; p: Ptr): OSErr;
;     EXTERNAL;
;

                MODULE  'module2'
                XDEF    ServiceDAQ
ServiceDAQ

                LINK    A6,#locals
                MOVEM.L D2-D7/A2-A4,-(SP)
                MOVEA.L p(A6),A4            ; Get address of data array.
                MOVE.L  count(A6),D7        ; Get total points to acquire.
                CLR.L   D6                  ; Clear point counter.
                MOVE.W  slot(A6),D1         ; Get slot number,
                EXT.L   D1                  ; clear upper word,
                LSL.L     #8,D1                 ; and multiply slot by $100000,
                LSL.L     #8,D1
                LSL.L     #4,D1
                MOVEA.L   #$800000,A3           ; and add $800000.
                ADDA.L    D1,A3                 ; A3 contains base address of slot

@1
                MOVE.W    (A3),D1               ; Fetch contents of status register
                BTST.L    #cAvail,D1            ; Is Conv Avail bit set?
                BEQ.S     @2
                MOVE.W    FIFO(A3),D0           ; Yes, fetch the result.
                EXT.L     D0
                MOVE.L    D0,(A4)+
                ADDQ.L    #1,D6                 ; and add 1 to the counter.
@2
                MOVE.W    D1,D0                 ; Check overrun and overflow flags
                ANDI.W    #error,D0
                BNE.S     @3                    ; Quit if error has occurred.
                CMP.L     D6,D7                 ; Check count
                BGT.S     @1                    ; Not finished, get another point.

@3
                CLR.W     func(A6)              ; Return ServiceDAQ=0.
                TST.W     D0                    ; Did I exit loop because of error?
                BEQ.S     @4
                MOVE.W    #10,func(A6)          ; Yes, return ServiceDAQ=10.
@4
                MOVEM.L   (SP)+,D2-D7/A2-A4
                UNLK      A6
                MOVEA.L   (SP)+,A0
                ADDA.L    #10,SP
                JMP       (A0)

;
;   FUNCTION AddDAQ(slot: INTEGER; count: LONGINT; p: Ptr): OSErr;
;      EXTERNAL;
;

                MODULE    'module3'
                XDEF      AddDAQ
AddDAQ

                LINK      A6,#locals
                MOVEM.L   D2-D7/A2-A4,-(SP)
                MOVEA.L   p(A6),A4              ; Get address of data array.
                MOVE.L    count(A6),D7          ; Get total points to acquire.
                CLR.L     D6                    ; Clear point counter.
                MOVE.W    slot(A6),D1           ; Get slot number,
                EXT.L     D1                    ; clear upper word,
                LSL.L     #8,D1                 ; and multiply slot by $100000,
                LSL.L     #8,D1
                LSL.L     #4,D1
                MOVEA.L   #$800000,A3           ; and add $800000.
                ADDA.L    D1,A3                 ; A3 contains base address of slot

@1
                MOVE.W    (A3),D1               ; Fetch contents of status register
                BTST.L    #cAvail,D1            ; Is Conv Avail bit set?
                BEQ.S     @2
                MOVE.W    FIFO(A3),D0           ; Yes, fetch the result.
                EXT.L     D0
                ADD.L     (A4),D0
                MOVE.L    D0,(A4)+
                ADDQ.L    #1,D6                 ; and add 1 to the counter.
```

```
@2
                MOVE.W    D1,D0                  ; Check overrun and overflow flags
                ANDI.W    #error,D0
                BNE.S     @3                     ; Quit if error has occurred.
                CMP.L     D6,D7                  ; Check count
                BGT.S     @1                     ; Not finished, get another point.

@3
                CLR.W     func(A6)               ; Return AddDAQ=0.
                TST.W     D0                     ; Did I exit loop because of error?
                BEQ.S     @4
                MOVE.W    #10,func(A6)           ; Yes, return AddDAQ=10.
@4
                MOVEM.L   (SP)+,D2-D7/A2-A4
                UNLK      A6
                MOVEA.L   (SP)+,A0
                ADDA.L    #10,SP
                JMP       (A0)

;
;  FUNCTION AddDAQ2(slot: INTEGER; count: LONGINT; p: Ptr): OSErr;
;     EXTERNAL;
;

                MODULE    'module4'
                XDEF      AddDAQ2
AddDAQ2

                LINK      A6,#locals
                MOVEM.L   D2-D7/A2-A4,-(SP)
                MOVEA.L   p(A6),A4               ; Get address of data array.
                MOVE.L    count(A6),D7           ; Get total points to acquire.
                CLR.L     D6                     ; Clear point counter.
                MOVE.W    slot(A6),D1            ; Get slot number,
                EXT.L     D1                     ; clear upper word,
                LSL.L     #8,D1                  ; and multiply slot by $100000,
                LSL.L     #8,D1
                LSL.L     #4,D1
                MOVEA.L   #$800000,A3            ; and add $800000.
                ADDA.L    D1,A3                  ; A3 contains base address of slot

@1
                MOVE.W    (A3),D1                ; Fetch contents of status register
                BTST.L    #cAvail,D1             ; Is Conv Avail bit set?
                BEQ.S     @2
                MOVE.W    FIFO(A3),D0            ; Yes, fetch the result.
                EXT.L     D0
                ADD.L     -(A4),D0               ; go backwards, pre-decrement
                MOVE.L    D0,(A4)
                ADDQ.L    #1,D6                  ; and add 1 to the counter.
@2
                MOVE.W    D1,D0                  ; Check overrun and overflow flags
                ANDI.W    #error,D0
                BNE.S     @3                     ; Quit if error has occurred.
                CMP.L     D6,D7                  ; Check count
                BGT.S     @1                     ; Not finished, get another point.

@3
                CLR.W     func(A6)               ; Return AddDAQ2=0.
                TST.W     D0                     ; Did I exit loop because of error?
                BEQ.S     @4
                MOVE.W    #10,func(A6)           ; Yes, return AddDAQ2=10.
@4
                MOVEM.L   (SP)+,D2-D7/A2-A4
```

```
UNLK      A6
MOVEA.L   (SP)+,A0
ADDA.L    #10,SP
JMP       (A0)

END
```

Appendix C

Data acquisition desk accessory for the Macintosh.

File: Acquire.pas

```
PROGRAM AcquireDA;

{-----------------------------------------------------------------------

   Desk accessory to acquire analog data from a National Instruments
   NB-MIO-16 ADC card and place the results on the clipboard as ASCII
   text.  Based on TML desk accessory shell.

   Version 1.20 - Greg King, September 11, 1988.

   Compile using TML Pascal version 2.01 or later using the
   "Create Desk Accessory" option.

-----------------------------------------------------------------------}

{$C 'DRVR' 27 'AcquireDA'}    { Compile and link a desk accessory. }
{$T 'DFIL' 'DMOV'}            { Make a Font/DA Mover document. }
{$L Acquire.rsrc}             { Resources are here. }
{$O-}
{$R-}

USES Macintf,Library1,National;

FUNCTION FindADC: INTEGER;
VAR
   myBlock: spBlock;
   found: BOOLEAN;
   err: OSErr;
BEGIN
FindADC:=0;
WITH myBlock DO
   BEGIN
   spSlot:=9;
   spID:=0;
   spCategory:=1;
   spCType:=0;
   spDrvrSW:=0;
   spDrvrHW:=0;
   spHWDev:=0;
   spTBMask:=0;
   found:=FALSE;
   REPEAT
      err:=sNextTypesRsrc(@myBlock);
      IF err=noErr THEN
         found:=IsADC(@myBlock);
   UNTIL found OR (err<>noErr);
   IF found THEN
      FindADC:=spSlot-8;
   END;
END; {of FindADC}
```

```
PROCEDURE DoIt(VAR Device: DCtlEntry);
VAR
   s: Str255;
   data,scrap: Handle;
   watch: CursHandle;
   p: ^LONGINT;
   myDlg: DialogPtr;
   slot,interval,channel,count,k,n: LONGINT;
   gain,time,i,ResourceID: INTEGER;
   xtrigger,xgate,xclock,xsense,xlevel,bipolar: BOOLEAN;
   error: OSErr;
BEGIN
myDlg:=WindowPtr(Device.DCtlWindow);
GetDNum(myDlg,3,slot);
GetDNum(myDlg,5,count);
GetDNum(myDlg,7,channel);
GetDNum(myDlg,21,interval);
GetDNum(myDlg,29,n);
gain:=GetRadio(myDlg,9,12)-9;
time:=GetRadio(myDlg,14,19)-14;
xtrigger:=GetCheckBox(myDlg,25);
xclock:=GetCheckBox(myDlg,24);
xgate:=GetCheckBox(myDlg,23);
bipolar:=GetCheckBox(myDlg,26);
xsense:=GetCheckBox(myDlg,27);
xlevel:=GetCheckBox(myDlg,28);
IF CheckSlot(slot) THEN
   BEGIN
   data:=NewHandle(count*SIZEOF(LONGINT));
   IF MemError=noErr THEN
      BEGIN
      watch:=GetCursor(WatchCursor);
      SetCursor(watch^^);
      HLock(data);
      InitMIO16(slot);
      k:=1;
      error:=noErr;
      WHILE (k<=n) AND (error=noErr) DO
         BEGIN
         error:=DAQ(slot,channel,gain,time,interval,count,xtrigger,xgate,xclock,
                    xsense,xlevel,bipolar,k<>1,data^);
         k:=k+1;
         END;
      IF error=noErr THEN
         BEGIN
         p:=POINTER(data^);
         scrap:=NewHandle(0);
         i:=1;
         error:=noErr;
         WHILE (i<=count) AND (error=noErr) DO
            BEGIN
            NumToString(p^,s);
            error:=StrAndHand(CONCAT(s,CHR(13)),scrap);
            p:=POINTER(ORD4(p)+SIZEOF(LONGINT));
            i:=i+1;
            END;
         IF error=noErr THEN
            BEGIN
            k:=ZeroScrap;
            HLock(scrap);
            k:=PutScrap(GetHandleSize(scrap),'TEXT',scrap^);
            HUnLock(scrap);
            END
         ELSE
            BEGIN
            SysBeep(3);
            SysBeep(3);
            SysBeep(3);
            END;
         DisposHandle(scrap);
         END
      ELSE
         SysBeep(3);
      HUnLock(data);
      DisposHandle(data);
      InitCursor;
      END
   ELSE
      BEGIN
      SysBeep(3);
      SysBeep(3);
      END;
   END
ELSE
   BEGIN
   ResourceID:= $BFE0+32*(-Device.DCtlRefNum);
   i:=StopAlert(ResourceID+2,NIL);
   END;
END; {of DoIt}

PROCEDURE UpdateDA(VAR Device: DCtlEntry);     { Redraw the dialog box in }
BEGIN                                          { response to an update    }
   BeginUpdate(WindowPtr(Device.DCtlWindow)); { event.                   }
   DrawDialog(DialogPtr(Device.DCtlWindow));
   EndUpdate(WindowPtr(Device.DCtlWindow))
END; { of UpdateDA }

{ ***** The Open, Ctl, and Close PROCEDUREs for a Desk Accessory *****
  These may have to be renamed for other compilers.                   }


PROCEDURE Close(VAR Device: DCtlEntry;
                VAR Block:  ParamBlockRec);

{  Well this is it!  The Desk Manager has called you to tell you
   this is the END.  You are not allowed to do anything but to
   remove your windows, menu, etc. and reclaim any storage.      }

BEGIN
WITH Device DO
   BEGIN
   DisposDialog(DialogPtr(Device.DCtlWindow));
   dctlWindow := NIL;
   END;
END; { of Close }


PROCEDURE Open(VAR Device: DCtlEntry;
               VAR Block:  ParamBlockRec);

{  This procedure is the first call your Desk Accessory will receive from
   the system to open the Desk Accessory.  Note that it is possible to
   receive a call to Open even if we are already open.  If this is the
   first call to Open then allocate any required 'global' storage and
   perform appropriate actions for your desk accessory.                  }

VAR
```

```
    myWorld: SysEnvRec;
    ResourceID: INTEGER;
    TmpPtr:      Ptr;
    WPeek:       WindowPeek;
    theDialog: DialogPtr;
    i,j,k: INTEGER;
    err: OSErr;

BEGIN
{ Check to see if this is the first time to be Opened. }
IF Device.DctlWindow = NIL THEN
   BEGIN

   { Compute resource ID of this Desk Accessory.  Remember Font/DA Mover can
     change them on you. That's why we use resource numbers which are OWNED
     by a particular DRVR resource for Menus, Windows, etc.}
   ResourceID := $BFE0 + 32 * (-Device.DCtlRefNum);

   { Create a hole in the heap.  It is good practice to keep Window records
     off of the bottom of the Application Heap. }
   TmpPtr := NewPtr($1000);

   { Create the desk accessory's dialog window }
   theDialog := GetNewDialog(ResourceID,NIL,pointer(-1));
   Device.DctlWindow := pointer(theDialog);
   WPeek := WindowPeek(Device.DCtlWindow);
   WPeek^.WindowKind := Device.DCtlRefNum;

   SetCheckBox(theDialog,9,TRUE);
   SetCheckBox(theDialog,14,TRUE);
   EnableCtl(theDialog,28,GetCheckBox(theDialog,27));

   { Deallocate our temporary pointer }
   DisposPtr(TmpPtr);

   { check environment }
   err:=SysEnvirons(1,myWorld);
   IF (err=noErr) AND (myWorld.machineType=envMacII) THEN
      BEGIN
      i:=FindADC;
      IF i<>0 THEN
         BEGIN
         SetDNum(theDialog,3,i);
         SelIText(theDialog,3,0,255);
         END
      ELSE
         BEGIN
         j:=StopAlert(ResourceID+1,NIL);
         CloseDeskAcc(Device.DCtlRefNum);
         END;
      END
   ELSE
      BEGIN
      j:=StopAlert(ResourceID+3,NIL);
      CloseDeskAcc(Device.DCtlRefNum);
      END;
   END;
END; { of Open }

PROCEDURE Ctl(VAR Device: DCtlEntry;
              VAR Block:  ParamBlockRec);

VAR
   Trick: RECORD
```

```
             CASE INTEGER OF
                0: (CSParam: ARRAY[0..1] OF INTEGER);
                1: (EventPtr: ^EventRecord)
             END;
   anEvent: EventRecord;
   mydialog: DialogPtr;
   i: INTEGER;
   b: BOOLEAN;

BEGIN

{ Find out what happened. }
CASE Block.csCode OF
   accEvent:  BEGIN
              Trick.CSParam[0] := Block.CSParam[0];
              Trick.CSParam[1] := Block.CSParam[1];
              CASE Trick.EventPtr^.what OF
                 mouseDown,keyDown:
                    IF DialogSelect(Trick.EventPtr^,mydialog,i) THEN
                       IF mydialog=DialogPtr(Device.DCtlWindow) THEN
                          { process mouse event }
                          CASE i OF
                             1: DoIt(Device);
                             2: CloseDeskAcc(Device.DCtlRefNum);
                             9,10,11,12:
                                ChangeRadio(mydialog,9,12,i);
                             14,15,16,17,18,19:
                                ChangeRadio(mydialog,14,19,i);
                             23,24,25,26,28:
                                SetCheckBox(mydialog,i,
                                            NOT GetCheckBox(mydialog,i));
                             27: BEGIN
                                 SetCheckBox(mydialog,27,NOT
GetCheckBox(mydialog,27));
                                 EnableCtl(mydialog,28,GetCheckBox(mydialog,27));
                                 END;
                             END; {CASE i}
                {mouseUp:     ;} { Ignore }
                {keyUp:       ;} { Ignore }
                 updateEvt:
                    UpdateDA(Device);
                {activateEvt: ;} { No special action required. }
              END; {CASE Trick.EventPtr^.what}
              END;
  {accRun:    ;} { I don't use any of these events. }
   accCursor: BEGIN {blink text insertion point}
              anEvent.what:=0;
              b:=DialogSelect(anEvent,mydialog,i);
              END;
  {accMenu:   ;
   accUndo:   ;
   accCut:    ;
   accCopy:   ;
   accPaste:  ;
   accClear:  ;}
END; {CASE Block.csCode}

END; { of Ctl }

BEGIN
   { No main program allowed. }
END.
```

Appendix D

Assembly language FFT listing for Macintosh II.

```
;
;  File FFT.a
;
;  Stand-alone subroutine to calculate FFT kernal.
;
;  AsmFFT(xr,xi: xPtr; N: LONGINT)
;
;  Greg King
;  Jan 16/88
;
;  asm fft.a -l
;  link -c "????" -t "????" -rt fft1=0 ∂
;             -sn Main="AsmFFT" ∂
;             fft.a.o ∂
;             -o FFTDriver

             MACHINE    MC68020
             MC68881

StackFrame              RECORD            20,DECREMENT
xr           DS.L       1
xi           DS.L       1
N            DS.L       1
Return       DS.L       1
A6Link       DS.L       1
temp         DS.X       1
u            DS.X       1
v            DS.X       1
I            DS.L       1
JK           DS.L       1
JKL          DS.L       1
K            DS.L       1
L            DS.L       1
L2           DS.L       1
LL           DS.L       1
M            DS.L       1
NDL          DS.L       1
localSize    EQU        *
             ENDR

J            EQU        D6
DN           EQU        D7

tr           EQU        FP2
ti           EQU        FP3
t            EQU        FP4
tt           EQU        FP5
wr           EQU        FP6
wi           EQU        FP7

AsmFFT       PROC       EXPORT
             WITH       stackFrame
             LINK       A6,#localSize
             MOVEM.L    A2-A4/D2-D7,-(SP)
             MOVE.L     N(A6),DN          ; register D7 contains N
```

```
          MOVEA.L    xr(A6),A3        ; register A3 contains base address of array xr
          MOVEA.L    xi(A6),A4        ; register A4 contains base address of array xi
          LEA        temp(A6),A2      ; stuff address in register, I use it a lot
          FLOG2.L    DN,FP0
          FMOVE.L    FP0,M(A6)        ; m:=log2(N)
          MOVE.L     DN,D0            ; ndl:=N/2
          ASR.L      #1,D0
          MOVE.L     D0,NDL(A6)
          MOVE.L     #1,L(A6)         ; l:=1
          MOVE.L     #10,LL(A6)       ; ll:=l*10
          MOVE.L     #1,I(A6)         ; i:=1
LoopI
          FMOVECR.X  #$0,FP0          ; u:=2*pi*ndl/(-N)
          FMUL.W     #2,FP0
          FMUL.L     NDL(A6),FP0
          FDIV.L     DN,FP0
          FNEG.X     FP0
          FMOVE.X    FP0,u(A6)
          FMOVECR.X  #$F,FP0          ; v:=0.0
          FMOVE.X    FP0,v(A6)
          CLR.L      JK(A6)           ; jk:=0
          MOVE.L     LL(A6),JKL(A6)   ; jkl:=ll
          MOVE.L     L(A6),D0
          MOVE.L     D0,L2(A6)        ; l2:=l
          ASL.L      #1;D0            ; l:=l*2
          MOVE.L     D0,L(A6)
          MOVE.L     LL(A6),D0        ; ll:=ll*2
          ASL.L      #1,D0
          MOVE.L     D0,LL(A6)
          CLR.L      K(A6)            ; k:=0
LoopK
          FSINCOS.X  v(A6),wr:wi      ; wr:=COS(v) wi:=SIN(v)
          CLR.L      J                ; j:=0

; I don't maintain seperate pointers for xr1,xr2,xi1, and xi2 because
; I'm short on address registers.  Instead I use indexed addressing.

          MOVE.L     JK(A6),D4        ; offset of subarray
          MOVE.L     JKL(A6),D5       ; offset of 2nd half of subarray
LoopJ
          LEA        (A3,D5.L),A0     ; tr:=RealToReal96(xr2^)
          MOVEA.L    A2,A1
          BSR        ConvertTo
          MOVEA.L    A2,A1
          FMOVE.X    (A1),tr
          LEA        (A4,D5.L),A0     ; ti:=RealToReal96(xi2^)
          BSR        ConvertTo
          FMOVE.X    (A2),ti

          FMOVE.X    tr,t             ; t:=tr*wr-ti*wi
          FMUL.X     wr,t
          FMOVE.X    ti,FP0
          FMUL.X     wi,FP0
          FSUB.X     FP0,t

          FMOVE.X    tr,tt            ; tt:=tr*wi+ti*wr
          FMUL.X     wi,tt
          FMOVE.X    ti,FP0
          FMUL.X     wr,FP0
          FADD.X     FP0,tt

          LEA        (A3,D4.L),A0     ; tr:=RealToReal96(xr1^)
          MOVEA.L    A2,A1
          BSR        ConvertTo
```

```
        MOVEA.L   A2,A1
        FMOVE.X   (A1),tr
        LEA       (A4,D4.L),A0      ; ti:=RealToReal96(xi1^)
        BSR       ConvertTo
        FMOVE.X   (A2),ti

        FMOVE.X   tr,FP0            ; xr2^:=tr-t
        FSUB.X    t,FP0
        MOVEA.L   A2,A1
        FMOVE.X   FP0,(A1)
        LEA       (A3,D5.L),A0
        BSR       ConvertFrom

        FMOVE.X   ti,FP0            ; xi2^:=ti-tt
        FSUB.X    tt,FP0
        MOVEA.L   A2,A1
        FMOVE.X   FP0,(A1)
        LEA       (A4,D5.L),A0
        BSR       ConvertFrom

        FMOVE.X   tr,FP0            ; xr1^:=tr+t
        FADD.X    t,FP0
        MOVEA.L   A2,A1
        FMOVE.X   FP0,(A1)
        LEA       (A3,D4.L),A0
        BSR       ConvertFrom

        FMOVE.X   ti,FP0            ; xi1^:=ti+tt
        FADD.X    tt,FP0
        MOVEA.L   A2,A1
        FMOVE.X   FP0,(A1)
        LEA       (A4,D4.L),A0
        BSR       ConvertFrom

        ADD.L     LL(A6),D4         ; jk':=jk'+ll
        ADD.L     LL(A6),D5         ; jkl':=jkl'+ll
        ADD.L     L(A6),J           ; j:=j+l
        CMP.L     J,DN
        BGT       LoopJ

        FMOVE.X   v(A6),FP0         ; v:=v+u
        FADD.X    u(A6),FP0
        FMOVE.X   FP0,v(A6)
        ADDI.L    #10,JK(A6)        ; jk:=jk+10
        ADDI.L    #10,JKL(A6)       ; jkl:=jkl+10
        ADDI.L    #1,K(A6)          ; k:=k+1
        MOVE.L    L2(A6),D0
        CMP.L     K(A6),D0
        BGT       LoopK

        MOVE.L    NDL(A6),D0        ; ndl:=ndl/2
        ASR.L     #1,D0
        MOVE.L    D0,NDL(A6)
        ADDI.L    #1,I(A6)          ; i:=i+1
        MOVE.L    M(A6),D0
        CMP.L     I(A6),D0
        BGE       LoopI

        MOVEM.L   (SP)+,A2-A4/D2-D7
        UNLK      A6
        MOVEA.L   (SP)+,A0
        ADDA.L    #12,SP
        JMP       (A0)
```

```
            ENDWITH

;-----------------------------------------------------------------------------
; Auxiliary routines to convert to and from IEEE 80 bit extended and
; Motorola 96 bit extended.
;
; Upon entry register A0 contains the address of the 80 bit value and
; register A1 contains the address of the 96 bit value.  The contents
; of these two registers are destroyed.
;-----------------------------------------------------------------------------
ConvertTo
            MOVE.W      (A0)+,(A1)+         ; copy exponent
            CLR.W       (A1)+               ; put in pad
            MOVE.L      (A0)+,(A1)+         ; copy mantissa
            MOVE.L      (A0)+,(A1)+
            RTS

ConvertFrom
            MOVE.W      (A1)+,(A0)+         ; copy exponent
            ADDQ.L      #2,A1               ; skip pad
            MOVE.L      (A1)+,(A0)+         ; copy mantissa
            MOVE.L      (A1)+,(A0)+
            RTS

            ENDPROC
            END
```

Appendix E

ICP lines for 21 elements.

This appendix contains summary tables for each of the 21 elements currently contained in our database of ICP lines acquired using FTS. The elements are: C, Cd, Co, Cr, Cu, Fe, La, Mg, Mn, Mo, Ni, Pb, Si, Sn, Ta, Th, U, V, W, Zn, and Zr. The database contains all observable lines in the range from 185 – 315 nm.

The intensities are relative to the most intense line for each element. A relative intensity of 0 means that the value has not been calculated. This usually signifies a spectral overlap. The entry "NLV" denotes lines where no literature wavelengths have been reported, in the case of U and Th it is likely that the lines have already been compiled by the spectroscopy group at Los Alamos National Laboratory.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
201.1503	0.0337	Co	II	201.150
202.2342	0.0295	Co	II	202.234
202.7027	0.0201	Co	II	202.702
219.3604	0.0150	Co	II	219.361
221.1430	0.0142	Co	II	221.143
221.4784	0.0192	Co	II	221.479
223.2067	0.0182	Co	II	223.208
224.5117	0.1322	Co	I	224.513
225.6747	0.0181	Co		225.674
226.0006	0.0153	Co		226.001
227.6530	0.0160	Co	I	227.653
228.3512	0.0284	Co	II	228.351
228.6156	1	Co	II	228.616
229.1980	0.0534	Co		NLV
229.3374	0.0434	Co	II	229.339
229.9758	0.0182	Co	II	229.975
230.1392	0.0378	Co	II	230.140
230.7846	0.6997	Co	II	230.786
230.9002	0.0174	Co	I	230.902
231.1589	0.5974	Co	II	231.160
231.2569	0.0177	Co	II	231.255
231.3601	0.0231	Co	II	231.364
231.4053	0.4257	Co	II	231.405
231.4970	0.2498	Co	II	231.498
231.7065	0.0249	Co		231.705
232.4311	0.1282	Co	II	232.432
232.6123	0.1249	Co	II	232.613
232.6475	0.1473	Co	II	232.648
233.0345	0.0869	Co	II	233.035
233.6226	0.0616	Co	II	233.624
233.7898	0.0282	Co	I	233.793
234.1118	0.0146	Co		234.112
234.4268	0.0979	Co	II	234.426
234.4612	0.0152	Co		234.464
234.6549	0.0174	Co		234.660
234.7386	0.2010	Co	II	234.739
235.3420	0.3012	Co		235.343
235.8197	0.0131	Co	I	235.818
236.0506	0.0281	Co		236.050
236.0788	0.0142	Co		NLV
236.1514	0.0170	Co	II	236.153
236.3788	0.4160	Co	II	236.379
237.8620	0.3840	Co	II	237.862
238.1762	0.0421	Co	II	238.175
238.3445	0.2905	Co	II	238.345
238.6353	0.1955	Co	II	238.637
238.8909	0.6583	Co	II	238.892
238.9529	0.0951	Co	II	238.956
239.7374	0.1134	Co	I	239.739
240.4156	0.0660	Co	II	240.417
240.7240	0.1144	Co	I	240.725
240.7659	0.0652	Co	II	240.766
240.8400	0.0197	Co		240.841
240.8736	0.0724	Co	II	240.875
241.1635	0	Co	II	241.162

Table E-1. Co ICP emission lines, part 1.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
241.4064	0.0873	Co	II	241.406
241.4457	0.0566	Co	I	241.446
241.5287	0.0377	Co	I	241.530
241.6892	0.0378	Co	II	241.689
241.7652	0.1088	Co	II	241.765
242.4918	0.0618	Co	I	242.493
243.2199	0.0341	Co	I	243.221
243.6658	0.0210	Co	I	234.666
244.9987	0.0169	Co	II	245.000
246.4194	0.0139	Co	II	246.420
250.6444	0.0176	Co	II	250.646
251.9815	0.0209	Co	II	251.982
252.1346	0.0233	Co	I	252.136
252.4969	0.0262	Co	II	252.497
252.8602	0.0467	Co	II	252.862
252.8957	0.0163	Co	I	252.897
254.1946	0.0294	Co	II	254.194
255.9390	0.0344	Co	II	255.941
256.4026	0.0813	Co	II	256.404
258.0313	0.2338	Co	II	258.033
258.2225	0.0859	Co	II	258.224
258.7203	0.1047	Co	II	258.722
263.2244	0.0140	Co	II	263.224
266.3535	0	Co	II	266.353
269.4669	0.0352	Co	II	269.468

Table E-2. Co ICP emission lines, part 2.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
205.5590	0.2964	Cr	II	205.559
206.1567	0.2333	Cr	II	206.154
206.5500	0.1639	Cr	II	206.546
212.9865	0.0125	Cr	II	212.989
213.0193	0.0099	Cr	II	213.022
213.2636	0.0129	Cr	II	213.262
213.3470	0.0270	Cr	II	213.349
213.4524	0.0172	Cr	II	213.452
213.5414	0.0131	Cr	II	213.542
224.8282	0.0119	Cr	II	224.830
224.8550	0.0099	Cr	II	224.856
225.5999	0.0103	Cr	II	225.601
229.7159	0.0185	Cr	II	229.717
230.7178	0.0149	Cr	II	230.720
231.4711	0.0122	Cr	II	231.471
231.9358	0.0141	Cr	II	231.938
232.0059	0.0109	Cr	II	232.008
265.3566	0.0808	Cr	II	265.357
265.8567	0.0858	Cr	II	265.859
266.1706	0.0244	Cr	II	266.173
266.3399	0.1755	Cr	II	266.342
266.6001	0.2643	Cr	II	266.602
266.8694	0.1133	Cr	II	266.871
267.1789	0.1580	Cr	II	267.180
267.2822	0.1162	Cr	II	267.283
267.7153	1	Cr	II	267.713
267.8778	0.1874	Cr	II	267.879
268.7074	0.1003	Cr	II	268.709
268.8276	0.0178	Cr	II	268.829
269.1032	0.1391	Cr	II	269.103
269.3514	0.0150	Cr	II	269.353
269.7894	0.0240	Cr	II	269.790
269.8386	0.0948	Cr	II	269.840
269.8666	0.0942	Cr	II	269.868
270.3557	0.0270	Cr	II	270.356
270.3839	0.0140	Cr	II	270.385
270.8784	0.0127	Cr	II	270.878
271.2291	0.0553	Cr	II	271.230
271.7499	0.0170	Cr	II	271.750
272.2726	0.0439	Cr	II	272.274
272.3620	0.0117	Cr	II	272.364
272.4031	0.0114	Cr	II	272.404
272.7238	0.0157	Cr	II	272.725
274.0085	0.0233	Cr	II	274.009
274.2023	0.0575	Cr	II	274.202
274.3620	0.0965	Cr	II	274.363
274.8966	0.1587	Cr	II	274.898
275.0716	0.1907	Cr	II	275.072
275.1848	0.1446	Cr	II	275.185
275.7703	0.1255	Cr	II	275.772
275.9373	0.0209	Cr	II	275.940
275.9704	0.0110	Cr	II	275.972
276.2571	0.3045	Cr	II	276.258
276.6515	0.5159	Cr	II	276.655
278.0276	0.0112	Cr	II	278.030

Table E-3. Cr ICP emission lines, part 1.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
278.5673	0.0125	Cr	II	278.569
279.2143	0.0134	Cr	II	279.216
280.0741	0.0182	Cr	II	280.077
281.1983	0.0177	Cr	II	281.200
281.8346	0.0127	Cr	II	281.836
282.1990	0.0094	Cr	II	282.201
282.2355	0.0460	Cr	II	282.238
283.0445	0.0369	Cr	II	283.046
283.5612	0.7166	Cr	II	283.563
284.0000	0.0301	Cr	II	284.001
284.3232	0.4697	Cr	II	284.324
284.9816	0.2850	Cr	II	284.983
285.1345	0.0216	Cr	II	285.135
285.5664	0.1501	Cr	II	285.567
285.6746	0.0134	Cr	II	285.677
285.8892	0.0992	Cr	II	285.891
286.0923	0.0610	Cr	II	286.092
286.2562	0.1298	Cr	II	286.257
286.5094	0.1360	Cr	II	286.510
286.6718	0.1108	Cr	II	286.672
286.7074	0.0143	Cr	II	286.709
286.7629	0.0786	Cr	II	286.765
287.0408	0.0300	Cr	II	287.043
287.3471	0.0214	Cr	II	287.346
287.5981	0.0543	Cr	II	287.597
287.7957	0.0140	Cr	II	287.797

Table E-4. Cr ICP emission lines, part 2.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
199.9690	0.0662	Cu	II	200.035
202.5502	0.0218	Cu	II	202.547
203.5843	0.0608	Cu	II	203.584
203.7123	0.0594	Cu	II	203.712
204.3789	0	Cu	II	204.379
205.4979	0.0679	Cu	II	205.497
210.4788	0.0591	Cu	II	210.478
211.2098	0.0901	Cu	II	211.209
212.2975	0.1076	Cu	II	212.297
212.6033	0.1421	Cu	II	212.603
213.5975	0.8320	Cu	II	213.598
214.8980	0.1154	Cu	II	214.897
216.5090	0.1666	Cu	I	216.509
217.8945	0.2801	Cu	I	217.894
217.9414	0	Cu	II	217.940
218.1730	0.1666	Cu	I	218.172
218.9620	0.1287	Cu	II	218.962
219.2259	0.6214	Cu	II	219.226
219.9584	0.4002	Cu	I	219.958
221.0263	0.1352	Cu	II	221.026
221.4606	0.1560	Cu	I	221.458
221.5707	0.0271	Cu	I	221.565
221.8100	0.4521	Cu	II	221.810
222.5707	0.0977	Cu	I	221.570
222.7775	0.3830	Cu	I	222.778
222.8865	0.1634	Cu	II	222.886
223.0087	0.6842	Cu		223.008
224.2604	0.3170	Cu	II	224.261
224.7003	1	Cu	II	224.700
226.3139	0.0245	Cu	I	226.308
227.6256	0.0807	Cu	II	227.625
229.3869	0.0706	Cu	I	229.384
229.4351	0.0760	Cu	II	229.436
236.9880	0.0452	Cu	II	236.989
261.8384	0.0219	Cu	I	261.837

Table E-5. Cu ICP emission lines.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
216.6764	0.0177	Fe	I	216.677
217.8086	0.0039	Fe	I	217.809
224.9196	0	Fe	II	224.918
225.3106	0.0038	Fe	II	225.312
226.0080	0.0064	Fe	II	226.008
226.7593	0.0047	Fe	II	226.758
227.9909	0.0117	Fe		227.992
229.8152	0.0018	Fe	I	229.818
232.7389	0.0628	Fe	II	232.739
233.1301	0.0536	Fe	II	233.131
233.2783	0.1714	Fe	II	233.280
233.8003	0.0982	Fe	II	233.800
234.3486	0.3757	Fe	II	234.349
234.3950	0.0377	Fe	II	234.396
234.4268	0.0860	Fe	II	234.428
234.8290	0.1434	Fe	II	234.830
235.4449	0.0352	Fe	II	235.447
235.4876	0.0192	Fe	II	235.489
235.9096	0.0723	Fe	II	235.910
235.9995	0.0645	Fe	II	235.999
236.0291	0.0612	Fe	II	236.029
236.2011	0.0202	Fe	II	236.202
236.4813	0.1089	Fe	II	236.483
236.6594	0.0110	Fe	II	236.659
236.8581	0.0397	Fe	II	236.859
237.3719	0.0856	Fe	II	237.373
237.5173	0.0290	Fe	II	237.519
237.9275	0.0346	Fe		237.927
238.0750	0.0515	Fe	II	238.076
238.2021	0.8480	Fe	II	238.204
236.3048	0.0110	Fe	II	238.305
238.3226	0.0283	Fe	II	238.324
238.4377	0.0208	Fe	II	238.439
238.8620	0.1637	Fe	II	238.863
239.5408	0.0201	Fe	II	239.541
239.5616	0.5352	Fe	II	239.562
239.9233	0.1527	Fe	II	239.924
240.4421	0.0196	Fe	II	240.443
240.4881	0.2776	Fe	II	240.488
240.6653	0.1146	Fe	II	240.666
241.0514	0.1514	Fe	II	241.052
241.1061	0.0770	Fe	II	241.107
241.3305	0.0626	Fe	II	241.331
248.2128	0	Fe	II	248.211
252.2834	0.0186	Fe	I	252.285
252.5369	0.0108	Fe	II	252.539
253.3607	0.0110	Fe	II	253.363
256.2521	0.0742	Fe		256.253
256.3471	0.0361	Fe	II	256.347
256.6882	0.0129	Fe	II	256.691
257.7908	0.0187	Fe	II	257.792
258.2595	0	Fe	II	258.258
258.5864	0.2574	Fe	II	258.588
259.1535	0.0284	Fe	II	259.154
259.8355	0.3292	Fe	II	259.837

Table E-6. Fe ICP emission lines, part 1.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
259.9381	1	Fe	II	259.940
260.7077	0.2837	Fe	II	260.709
261.1854	0.4176	Fe	II	261.187
261.3813	0.1740	Fe	II	261.382
261.7608	0.1196	Fe	II	261.762
262.1662	0.0431	Fe	II	262.167
262.5645	0.1538	Fe	II	262.567
262.8274	0.1501	Fe	II	262.829
263.1025	0.2180	Fe	II	263.105
263.1309	0.2331	Fe	II	263.132
271.4404	0.0548	Fe	II	271.441
271.9013	0.0424	Fe	I	271.902
272.0886	0.0204	Fe		272.090
272.7525	0.0602	Fe	II	272.754
273.9524	0.2793	Fe	II	273.955
274.3184	0.1089	Fe	II	274.320
274.6472	0.1774	Fe	II	274.648
274.6963	0.1510	Fe		274.698
274.9312	0.2484	Fe	II	274.932
275.5724	0.3353	Fe	II	275.574

Table E-7. Fe ICP emission lines, part 2.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
195.1642	0.0124	La		NLV
216.3655	0.0169	La	II	216.366
218.7864	0.0698	La	II	218.787
221.6062	0.0406	La	III	221.611
223.0723	0.0120	La		NLV
225.6747	0.3755	La	II	225.676
229.7732	0.3315	La	II	229.777
231.7816	0.0509	La		NLV
231.9423	0.0812	La		NLV
232.8735	0.0464	La		NLV
237.9370	0.1906	La	II	237.938
239.9621	0.0248	La		NLV
251.9218	0.0329	La	II	251.922
253.3127	0.0150	La		NLV
256.0369	0.0487	La		NLV
256.1824	0.0157	La		NLV
258.0795	0.0115	La		NLV
258.2515	0.0136	La		NLV
259.6077	0.0375	La		NLV
261.0326	1	La	II	261.034
261.6303	0.0131	La		NLV
266.6155	0.0165	La		NLV
267.2891	0.0898	La	II	267.291
268.1480	0.0268	La		NLV
269.5457	0.0879	La	II	269.547
271.5417	0.0180	La		NLV
278.0220	0.0408	La		NLV
279.1504	0.0484	La	II	279.151
279.6360	0.0168	La		NLV
279.8530	0.0654	La	II	279.856
280.8346	0.4904	La	II	280.839
284.0487	0.0190	La		NLV
285.5880	0.0406	La	II	285.590
288.0616	0.0475	La	II	288.065
288.5125	0.0752	La	II	288.514
289.3054	0.0757	La	II	289.307
295.0477	0.0193	La	II	295.050

Table E-8. La ICP emission lines.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
202.0883	0.0011	Mn		NLV
202.8157	0.0009	Mn		NLV
203.1734	0.0016	Mn		NLV
203.9046	0.0023	Mn		NLV
205.0093	0.0013	Mn		NLV
221.3850	0.0009	Mn	I	221.382
222.1820	0.0015	Mn	I	222.184
230.4986	0.0057	Mn	II	230.500
232.0423	0.0007	Mn	II	232.045
254.3443	0.0027	Mn	II	254.345
255.8569	0.0016	Mn	II	255.858
256.3630	0.0060	Mn	II	256.365
257.3716	0.0094	Mn		NLV
257.6082	1	Mn	II	257.610
257.8501	0.0056	Mn		NLV
259.1292	0.0083	Mn		NLV
259.3707	0.9492	Mn	II	259.373
260.5668	0.6954	Mn	II	260.569
261.0178	0.0201	Mn	II	261.020
261.8120	0.0205	Mn	II	261.814
262.5595	0.0185	Mn	II	262.559
263.2328	0.0156	Mn	II	263.235
263.8155	0.0082	Mn	II	263.817
263.9817	0.0065	Mn	II	263.984
267.2564	0.0088	Mn	II	267.259
270.1673	0.0205	Mn	II	270.170
270.5709	0.0187	Mn	II	270.574
270.8430	0.0150	Mn	II	270.845
271.0306	0.0102	Mn	II	271.033
271.1600	0.0088	Mn	II	271.158
279.4796	0.0675	Mn	I	279.482
279.8247	0.0495	Mn	I	279.827
280.1063	0.0338	Mn	I	280.108
293.3033	0.0287	Mn	II	293.306
293.9290	0.0436	Mn	II	293.930
294.9176	0.0486	Mn	II	294.921

Table E-9. Mn ICP emission lines.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
201.5100	0.1181	Mo	II	201.512
202.0303	0.3815	Mo	II	202.030
203.8444	0.2404	Mo	II	203.844
204.5966	0.2702	Mo	II	204.598
208.1673	0.0580	Mo	II	208.168
208.9518	0.0216	Mo	II	208.952
209.2512	0.0334	Mo	II	209.250
209.3103	0.0915	Mo	II	209.311
209.5282	0.0235	Mo	II	209.528
210.0843	0.0642	Mo	II	210.084
210.1695	0.0223	Mo	[illegible]	210.169
210.4286	0.0321	Mo	II	210.429
210.5012	0.0414	Mo	II	210.502
210.8030	0.0384	Mo	II	210.804
210.9939	0.0225	Mo	II	210.994
217.2894	0.0336	Mo	I	217.290
218.4350	0.0312	Mo	II	218.436
218.8950	0.0272	Mo	II	218.895
218.9389	0.0662	Mo	II	218.940
219.7463	0.0659	Mo	II	219.748
219.8138	0.0218	Mo	II	219.814
222.0975	0.0854	Mo	II	222.099
223.9407	0.0255	Mo	II	223.942
224.9306	0.0337	Mo	II	224.932
226.9691	0.1121	Mo	II	226.969
228.9879	0.0287	Mo	II	228.989
229.0297	0.0436	Mo	II	229.031
230.5652	0.0270	Mo	II	230.567
230.6973	0.0748	Mo	II	230.699
233.2114	0.0540	Mo	II	233.213
234.1581	0.0566	Mo	II	234.157
235.5438	0.0301	Mo	II	235.542
236.6094	0.0237	Mo	II	236.609
253.8445	0.0443	Mo	II	253.846
259.3691	0.0503	Mo	II	259.371
260.2790	0.0710	Mo	II	260.280
261.9327	0.0331	Mo	II	261.934
263.6662	0.1131	Mo	II	263.667
263.8742	0.3618	Mo	II	263.876
264.4328	0.2685	Mo	II	264.434
264.6471	0.1868	Mo	II	264.649
265.3328	0.1636	Mo	II	265.335
266.0562	0.3457	Mo	II	266.058
267.2822	0.3748	Mo	II	267.284
267.3253	0.1551	Mo	II	267.327
268.1341	0.0400	Mo	II	268.137
268.3215	0.2859	Mo	II	268.323
268.4118	0.5712	Mo	II	268.414
268.7980	0.3237	Mo	II	268.799
270.1409	0.2215	Mo	II	270.141
271.3480	0.0482	Mo	II	271.348
271.7338	0.1341	Mo	II	271.736
272.9661	0.0480	Mo	II	272.968
273.0182	0.0247	Mo	II	273.021
273.2863	0.1120	Mo	II	273.288

Table E-10. Mo ICP emission lines, part 1.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
273.6938	0.0646	Mo	II	273.696
274.3166	0.0655	Mo	I	274.307
274.6272	0.0989	Mo	II	274.630
275.0023	0.0249	Mo	II	275.003
275.6054	0.0399	Mo	II	275.607
276.3602	0.0794	Mo	II	276.363
276.9748	0.0430	Mo	II	276.976
277.4380	0.0565	Mo	II	277.440
277.5383	0.6800	Mo	II	277.540
278.0015	0.4096	Mo	II	278.002
278.4962	0.1296	Mo	II	278.499
280.7738	0.1476	Mo	II	280.775
281.6146	1	Mo	II	281.618
281.7427	0.0216	Mo	II	281.744
284.2122	0.0378	Mo	II	284.215
284.8211	0.5419	Mo	II	284.824
285.3208	0.0914	Mo	II	285.322
286.3788	0.0676	Mo	II	286.380
286.6678	0.0321	Mo	II	286.669
287.1501	0.2847	Mo	II	287.151
289.0977	0.1487	Mo	II	289.099
289.4427	0.0980	Mo	II	289.445
290.3038	0.0264	Mo	II	290.306
290.9106	0.0509	Mo	II	290.911
291.1904	0.1107	Mo	II	291.192
292.3378	0.1011	Mo	II	292.339
293.0484	0.0803	Mo	II	293.048
293.4278	0.0489	Mo	II	293.429

Table E-11. Mo ICP emission lines, part 2.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
212.8564	0.0222	Ni	II	212.857
213.8575	0.0180	Ni	II	213.860
215.8318	0.0068	Ni	I	215.831
215.8745	0.0226	Ni	II	215.873
216.1209	0.0197	Ni	II	216.121
216.5542	0.4315	Ni	II	216.556
216.9087	0.2000	Ni	II	216.910
217.4660	0.2417	Ni	II	217.467
217.5139	0.1559	Ni	II	217.516
217.7354	0.0042	Ni	II	217.736
217.9345	0.0178	Ni	II	217.935
218.0468	0.0214	Ni	II	218.046
218.4591	0.1558	Ni	II	218.461
218.5501	0.0376	Ni	II	218.551
218.8061	0.0077	Ni	II	218.804
220.1394	0.1215	Ni	II	220.141
220.6700	0.2040	Ni	II	220.670
221.0369	0.0895	Ni	II	221.038
221.6465	1	Ni	II	221.647
221.8255	0.0125	Ni		NLV
222.0392	0.0638	Ni	II	222.040
222.2939	0.2569	Ni	II	222.295
222.4346	0.0201	Ni	II	222.435
222.4859	0.2790	Ni	II	222.488
222.6316	0.1495	Ni	II	222.634
224.7246	0.0111	Ni	II	224.723
225.3841	0.2215	Ni	II	225.386
225.6146	0.0030	Ni	II	225.615
226.4449	0.2583	Ni	II	226.446
227.0200	0.3902	Ni	II	227.021
227.4707	0.0110	Ni	II	227.475
227.5681	0.0148	Ni	II	227.568
227.6455	0.0122	Ni	II	227.644
227.7268	0.0501	Ni	II	227.731
227.8306	0.0157	Ni		NLV
227.8757	0.1919	Ni	II	227.877
228.7076	0.1086	Ni	II	228.708
228.7644	0.0348	Ni	II	228.765
228.9980	0.0723	Ni	I	228.998
229.3120	0.0132	Ni	I	229.311
229.6536	0.1933	Ni	II	229.656
229.7134	0.2477	Ni	II	229.713
229.7478	0.1285	Ni	II	229.749
229.8254	0.1449	Ni	II	229.828
229.9643	0.0294	Ni	II	229.965
230.0090	0.0560	Ni	II	230.010
230.0766	0.0219	Ni	I	230.077
230.2478	0.0172	Ni	II	230.248
230.2989	0.4403	Ni	II	230.298
230.5255	0.0040	Ni	II	230.524
231.0945	0.2064	Ni	I	231.097
231.2337	0.1299	Ni	I	231.234
231.2904	0.0311	Ni	II	231.291
231.3640	0.0875	Ni		231.365
231.3975	0.0799	Ni	I	231.398

Table E-12. Ni ICP emission lines, part 1.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
231.6030	0.7217	Ni	II	231.603
231.7156	0.0731	Ni	I	231.716
231.8503	0.0046	Ni	II	231.848
232.0020	0.2765	Ni	I	232.003
232.1371	0.1175	Ni	I	232.138
232.2684	0.0193	Ni		232.268
232.4689	0.0086	Ni	I	232.464
232.5784	0.1066	Ni	I	232.579
232.6449	0.0272	Ni	II	232.644
232.9952	0.0417	Ni	I	232.996
233.4582	0.0713	Ni	II	233.456
233.6686	0.0048	Ni	II	233.670
233.7476	0.0192	Ni	I	233.749
233.7793	0.0054	Ni	I	233.782
234.1197	0.0193	Ni	II	234.118
234.3499	0.0060	Ni	II	234.348
234.5262	0.0212	Ni	II	234.526
234.5527	0.0774	Ni	I	234.555
234.6629	0.0063	Ni	I	234.663
234.7532	0.0139	Ni	I	234.752
235.6388	0.0151	Ni	II	235.641
236.2079	0.0038	Ni	I	236.206
236.6553	0.0097	Ni	II	236.656
236.7378	0.0156	Ni	II	236.740
237.5418	0.0384	Ni	II	237.542
238.2049	0.0073	Ni		NLV
239.4509	0.1466	Ni	II	239.452
240.1859	0.0123	Ni	I	240.184
241.3039	0.0115	Ni	II	241.305
241.6131	0.0808	Ni	II	241.614
241.9288	0.0086	Ni	I	241.931
243.3541	0.0037	Ni	II	243.357
243.7875	0.0243	Ni	II	243.789
247.3129	0.0123	Ni	II	243.315
250.5823	0.0113	Ni	II	250.584
251.0860	0.0298	Ni	II	251.087
254.5878	0.0159	Ni	II	254.590

Table E-13. Ni ICP emission lines, part 2.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
205.3300	0.0032	Pb	I	205.327
216.9995	0.2178	Pb	I	216.999
220.3522	1	Pb	II	220.351
223.7450	0.0059	Pb	I	223.743
224.6893	0.0203	Pb	I	224.689
239.3790	0.0125	Pb	I	239.379
240.1970	0.0164	Pb	I	240.195
241.1748	0.0066	Pb	I	241.174
244.3866	0.0097	Pb	I	244.384
244.6172	0.0180	Pb	I	244.619
247.6377	0.0499	Pb	I	247.638
257.7251	0.0152	Pb	I	257.726
261.3649	0.0108	Pb	I	261.365
261.4159	0.2337	Pb	I	261.418
266.3142	0.0806	Pb	I	266.317
280.1971	0.1763	Pb	I	280.200
282.3181	0.0166	Pb	I	282.319
283.3036	0.1443	Pb	I	283.307
287.3291	0.0184	Pb	I	287.332

Table E-14. Pb ICP emission lines.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
189.9242	0.1794	Sn	II	189.989
209.1583	0.0602	Sn	I	209.158
211.3929	0.1204	Sn	I	211.393
215.0830	0.0624	Sn		215.084
215.1499	0.2069	Sn		NLV
219.4486	0.1223	Sn	I	219.449
219.9339	0.2043	Sn	I	219.934
220.9651	0.3842	Sn	I	220.966
224.6041	0.6920	Sn	I	224.605
226.6008	0.0645	Sn	I	226.604
226.8921	0.6084	Sn	I	226.891
228.6672	0.0984	Sn	I	228.668
231.7233	0.2082	Sn	I	231.723
233.4805	0.2123	Sn	I	233.480
235.4836	0.8662	Sn	I	235.484
236.8216	0.0636	Sn		NLV
242.1690	0.2475	Sn	I	242.170
242.9478	0.4228	Sn	I	242.949
254.6535	0.0902	Sn	I	254.655
257.1578	0.0923	Sn	I	257.158
266.1228	0.0939	Sn	I	266.124
270.6486	0.8663	Sn	I	270.651
283.9961	1	Sn	I	283.999
286.3294	0.3140	Sn	I	286.332

Table E-15. Sn ICP emission lines.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
204.7411	0.0504	Ta		NLV
205.9070	0.1016	Ta		NLV
206.1947	0.0286	Ta		NLV
207.5537	0.0588	Ta		NLV
207.6327	0.0626	Ta		NLV
208.7278	0.0279	Ta		NLV
208.8676	0.0465	Ta		NLV
209.0613	0.0258	Ta		NLV
209.1341	0.0746	Ta		NLV
209.6987	0.0267	Ta		NLV
210.0343	0.0576	Ta		NLV
211.1206	0.0254	Ta		NLV
211.1819	0.0313	Ta		NLV
211.2841	0.0294	Ta		NLV
211.3702	0.0788	Ta		NLV
211.8345	0.0388	Ta		NLV
211.9936	0.0250	Ta		NLV
212.0912	0.0615	Ta		NLV
212.4824	0.0976	Ta		NLV
212.9220	0.0358	Ta		NLV
213.1792	0.1306	Ta		NLV
213.2899	0.0321	Ta		NLV
214.0142	0.2509	Ta	II	214.015
214.1512	0.0314	Ta		NLV
214.2519	0.0324	Ta		NLV
214.6877	0.2569	Ta	II	214.687
215.0629	0.1174	Ta		NLV
215.1399	0.1171	Ta		NLV
215.2717	0.0499	Ta		NLV
215.4193	0.0484	Ta		NLV
215.4663	0.0241	Ta		NLV
215.5088	0.0243	Ta		NLV
216.5022	0.0505	Ta		NLV
217.2097	0.0247	Ta		NLV
217.8018	0.0875	Ta		NLV
217.9540	0.1266	Ta		NLV
218.0675	0.0272	Ta		NLV
218.0881	0.0274	Ta		NLV
218.1787	0.0605	Ta		NLV
218.2706	0.3856	Ta	II	218.272
218.4684	0.0298	Ta		NLV
218.8488	0.0421	Ta		NLV
219.3221	0.1396	Ta		NLV
219.3906	0.1539	Ta		NLV
219.6055	0.4026	Ta	II	219.605
219.9689	0.2278	Ta	II	219.967
220.1172	0.0882	Ta		NLV
220.7158	0.0938	Ta		NLV
221.0051	0.1249	Ta	II	221.003
221.2421	0.0688	Ta		NLV
221.4358	0.0469	Ta		NLV
221.5600	0.2239	Ta		NLV
221.7839	0.0761	Ta		NLV
221.9406	0.0566	Ta		NLV
222.7847	0.4305	Ta	II	222.785

Table E-16. Ta ICP emission lines, part 1.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
223.6292	0.0605	Ta		NLV
223.7257	0.0496	Ta		NLV
223.9008	0.0897	Ta		NLV
223.9504	0.4367	Ta	II	223.948
224.3878	0.0638	Ta		NLV
224.8489	0.0946	Ta		NLV
224.9782	0.2322	Ta		NLV
225.0758	0.6639	Ta	II	225.076
225.3265	0.0336	Ta		NLV
225.4870	0.0605	Ta		NLV
225.5778	0.0997	Ta		NLV
225.6514	0.1189	Ta		NLV
225.8726	0.1159	Ta		NLV
225.9563	0.0917	Ta		NLV
226.0240	0.0339	Ta		NLV
226.0561	0.0319	Ta		NLV
226.1423	0.6469	Ta		NLV
226.2299	0.7379	Ta	II	226.230
226.9206	0.0625	Ta		NLV
226.9554	0.0762	Ta		NLV
227.0896	0.0354	Ta		NLV
227.1854	0.2674	Ta		NLV
227.2588	0.5709	Ta		NLV
227.3722	0.0597	Ta		NLV
227.5644	0.0415	Ta		NLV
228.1352	0.0275	Ta		NLV
228.1540	0.0396	Ta		NLV
228.2193	0.0863	Ta		NLV
228.3210	0.0540	Ta		NLV
228.5009	0.0905	Ta		NLV
228.5261	0.2691	Ta		NLV
228.6597	0.0287	Ta		NLV
228.7278	0.0811	Ta		NLV
228.8868	0.0249	Ta		NLV
228.9159	0.4910	Ta	II	228.916
229.2537	0.0946	Ta		NLV
229.5189	0.0251	Ta		NLV
229.8942	0.0326	Ta		NLV
229.9656	0.0357	Ta		NLV
230.1468	0.0619	Ta		NLV
230.2235	0.1859	Ta		NLV
230.2925	0.4716	Ta		NLV
230.3488	0.2866	Ta		NLV
230.8501	0.0505	Ta		NLV
231.2607	0.1850	Ta		NLV
231.5448	0.1859	Ta		NLV
231.9112	0.0769	Ta		NLV
233.1275	0.0371	Ta		NLV
233.2009	0.4664	Ta	II	233.198
233.3872	0.0268	Ta		NLV
233.4135	0.0454	Ta		NLV
233.4897	0.0724	Ta		NLV
233.5752	0.0417	Ta		NLV
233.8293	0.0922	Ta		NLV
234.0022	0.0396	Ta		0

Table E-17. Ta ICP emission lines, part 2.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
234.0933	0.0913	Ta		NLV
234.1607	0.0583	Ta		NLV
234.2520	0.0413	Ta		NLV
235.3861	0.0523	Ta		NLV
235.6054	0.0923	Ta		NLV
235.6911	0.0353	Ta		NLV
235.9163	0.0467	Ta		NLV
236.4260	0.2904	Ta		NLV
237.0761	0.0911	Ta		NLV
237.8320	0.0495	Ta		NLV
238.1133	0.2003	Ta	II	238.114
238.1556	0.0810	Ta		NLV
238.3733	0.0629	Ta		NLV
238.4295	0.0797	Ta		NLV
238.7081	0.6656	Ta	II	238.706
240.0649	0.6971	Ta	II	240.062
240.2123	0.0883	Ta		NLV
241.6878	0.1128	Ta	II	241.689
243.2713	0.1337	Ta	II	243.270
247.0904	0.0393	Ta	II	247.090
248.8696	0.0670	Ta	II	248.870
253.2107	0.0963	Ta	II	253.213
254.5488	0.0525	Ta	II	254.550
255.4613	0.1621	Ta	II	255.462
255.4912	0.0404	Ta	II	255.491
257.7379	0.0782	Ta	II	257.737
258.4027	0.0949	Ta	II	258.403
259.3658	0.1374	Ta	II	259.366
259.4242	0.1264	Ta		NLV
259.5574	0.0673	Ta	II	259.559
259.6435	0.1765	Ta	II	259.645
260.3476	0.3995	Ta	II	260.349
261.2595	0.0834	Ta	II	261.261
263.5573	1	Ta	II	263.559
265.1208	0.2058	Ta	II	265.122
265.8840	0.1324	Ta	II	265.886
266.5590	0.1137	Ta	II	266.590
267.5892	0.5711	Ta	II	267.590
268.0042	0.2240	Ta	II	268.006
268.0648	0.1756	Ta	II	268.066
268.5160	0.6225	Ta	II	268.511
269.4511	0.6456	Ta	II	269.452
270.9262	0.2286	Ta	II	270.927
272.7417	0.3288	Ta	II	272.744
273.9252	0.0958	Ta	II	273.926
275.2469	0.4435	Ta	II	275.249
276.1651	0.3298	Ta	II	276.168
278.4943	0.0890	Ta	II	278.497
279.7737	0.3216	Ta	II	279.776
281.7083	0.1081	Ta		281.710
284.4441	0.1670	Ta	II	284.446

Table E-18. Ta ICP emission lines, part 3.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
203.7253	0.0265	Th		NLV
210.8362	0.0343	Th		NLV
211.1690	0.0380	Th		NLV
211.7534	0.0600	Th		NLV
212.1650	0.0527	Th		NLV
213.2757	0.0433	Th		NLV
217.1222	0.0285	Th		NLV
218.7599	0.0710	Th		NLV
221.0640	0.0386	Th		NLV
222.1094	0.0381	Th		NLV
222.4227	0.0273	Th		NLV
223.0351	0.1350	Th		NLV
223.9758	0.0573	Th		NLV
224.3271	0.0333	Th		NLV
226.3398	0.0285	Th		NLV
226.5908	0.0236	Th		NLV
228.0423	0.0999	Th		NLV
228.1967	0.0397	Th		NLV
230.5460	0.0291	Th		NLV
232.6919	0.1053	Th		NLV
237.7829	0.1379	Th		NLV
240.4156	0.0369	Th		NLV
242.3657	0.0274	Th		NLV
248.9622	0.0383	Th		NLV
252.0642	0.0227	Th		NLV
254.7880	0.0661	Th	II	254.790
256.1919	0.0428	Th		NLV
256.5580	0.2700	Th	II	256.560
256.6581	0.1573	Th	II	256.659
257.6674	0.1384	Th	II	257.669
257.9415	0.0527	Th		NLV
258.9044	0.1803	Th	II	258.906
259.7037	0.1824	Th	II	259.705
260.0864	0.1439	Th	II	260.089
260.9849	0.0970	Th		NLV
261.8897	0.1594	Th	II	261.890
262.3436	0.1901	Th	II	262.345
262.5728	0.2358	Th	II	262.574
269.2395	0.7219	Th	II	269.242
270.3945	0.2993	Th	II	270.396
272.1672	0.2001	Th	II	272.170
272.9320	0.3114	Th	II	272.933
273.2791	0.2635	Th	II	273.282
273.4394	0.1886	Th	II	273.442
274.3039	0.1733	Th	II	274.307
274.7145	0.8454	Th	II	274.716
274.9513	0.1706	Th		NLV
275.2158	0.5243	Th	II	275.217
276.0383	0.1302	Th	II	276.040
276.3584	0.1573	Th	II	276.362
276.5113	0.1272	Th		NLV
276.8823	0.4864	Th	II	276.885
277.0803	0.2164	Th	II	277.082
277.1488	0.1931	Th	II	277.149
278.6890	0.0874	Th		NLV

Table E-19. Th ICP emission lines, part 1.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
278.7114	0.1309	Th		NLV
279.4231	0.0807	Th	II	279.427
279.8247	0.0738	Th		NLV
280.7814	0.0918	Th		NLV
280.8974	0.1062	Th		NLV
282.2009	0.0797	Th	II	282.203
282.6837	0.1100	Th	II	282.686
283.2301	0.7039	Th	II	283.231
283.4469	0.0764	Th		NLV
283.6039	0.0686	Th		NLV
283.7290	1	Th	II	283.730
284.2804	0.1773	Th	II	284.281
285.1247	0.2145	Th	II	285.126
286.1397	0.0688	Th		NLV
287.0388	0.3040	Th	II	287.040
288.4262	0.1556	Th	II	288.429
288.5024	0.1964	Th	II	288.505
288.7796	0.1337	Th	II	288.781
289.1239	0.0653	Th	I	289.125
289.9709	0.1128	Th	II	289.972
292.5027	0.0785	Th	II	292.505
292.8228	0.0670	Th	II	292.825
294.2835	0.0931	Th	II	294.286
296.8661	0.0386	Th	II	296.869

Table E-20. Th ICP emission lines, part 2.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
213.1135	0	U		NLV
221.9275	0.0721	U		221.928
222.4597	0.0624	U		NLV
222.7069	0.0922	U		NLV
224.1404	0.0592	U		NLV
225.2935	0.1422	U		NLV
227.9960	0.0703	U		NLV
228.2771	0.0935	U		228.278
233.8729	0.1734	U		NLV
234.4612	0.1382	U		NLV
234.6987	0.1370	U		NLV
240.2234	0.2475	U		NLV
242.7487	0.0933	U		NLV
246.8301	0.2715	U		NLV
253.7281	0.0443	U		NLV
253.8414	0.0743	U		253.843
255.6172	0.1530	U	II	255.619
256.2774	0.0801	U		NLV
256.2933	0.1872	U		256.294
256.5390	0.3459	U		256.541
256.7105	0.0606	U		NLV
256.8806	0.0571	U		NLV
256.9698	0.1900	U		256.971
257.0653	0.0773	U		NLV
257.7315	0.0928	U		257.732
257.9142	0.0771	U		NLV
257.9559	0.0914	U		NLV
258.4397	0.1382	U		NLV
258.6186	0.0929	U		NLV
259.1228	0.2629	U		259.125
259.3545	0.1303	U		259.357
259.7671	0	U		259.769
260.6503	0.1810	U		NLV
260.6716	0.1828	U		NLV
260.8175	0.1118	U		260.820
262.4897	0.1893	U		NLV
262.5246	0.1410	U		NLV
262.8491	0.1538	U		NLV
262.8907	0.2866	U		262.893
263.2645	0.2440	U		263.266
263.2963	0.3518	U		263.298
263.5506	1	U		263.553
264.4109	0.1659	U		NLV
264.5458	0.5360	U		264.547
264.9057	0.2768	U	II	264.907
265.2802	0.2773	U		265.283
266.4134	0.2321	U		266.415
266.9209	0.3194	U		266.917
268.3267	0.5130	U		268.328
268.5960	0.3661	U		268.598
269.1014	0.4724	U		269.104
269.2342	0.3018	U		269.236
269.3776	0.2206	U		269.377
269.5474	0.2997	U		269.549
269.8034	0.3530	U		269.806

Table E-21. U ICP emission lines, part 1.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
270.6963	0.4961	U		270.695
272.3137	0.2326	U		NLV
273.3944	0.3927	U		273.397
274.4382	0.2912	U		274.440
275.4132	0.6996	U		275.416
275.5120	0.3070	U		275.513
276.2866	0.3084	U		276.286
276.5371	0.2767	U		276.540
277.0026	0.3288	U		277.004
278.4439	0.2644	U	II	278.445
279.3911	0.6131	U	II	279.394
279.5512	0.2596	U		NLV
280.2539	0.6154	U		280.256
280.7111	0.3350	U	II	280.705
280.8993	0.2157	U		280.898
281.1373	0.2017	U		281.135
281.7944	0.3488	U	II	281.796
282.1107	0.5506	U	II	282.112
282.4354	0.1379	U	II	282.437
282.8920	0.2778	U		282.89
283.2049	0.3245	U		283.206
283.9864	0.1478	U	II	283.969
285.8892	0.1997	U	II	285.890
286.5668	0.3581	U	II	286.568
288.2718	0.2621	U		288.274
288.9607	0.2999	U	II	288.963
289.4488	0.1017	U		289.451
290.6781	0.1643	U		290.680
290.8249	0.2020	U		290.828
294.1896	0.1656	U	II	294.192
295.6050	0.0738	U		295.606

Table E-22. U ICP emission lines, part 2.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
213.4096	0.0495	V	II	213.412
213.7285	0.0498	V	II	213.731
213.8145	0.0206	V	II	213.817
213.9799	0.0376	V	II	213.980
214.0053	0.1328	V	II	214.006
214.1965	0.0626	V	II	213.194
214.3039	0.0243	V	II	214.304
234.2123	0.0180	V	II	234.214
235.2165	0.0196	V	II	235.218
238.0914	0.0146	V	II	238.091
238.9680	0	V	II	238.970
252.7893	0.0348	V	II	252.790
252.8818	0.0303	V	II	252.883
254.9258	0.0179	V	II	254.927
263.[illegible]658	0.0542	V	II	263.067
264.2189	0.0433	V	II	264.220
264.5813	0.0378	V	II	264.584
264.9311	0.0162	V	II	264.937
265.5639	0.0157	V	II	265.568
266.3193	0.0223	V	II	266.325
267.1978	0.2004	V	II	267.200
267.7775	0.2326	V	II	267.780
267.8553	0.1502	V	II	267.857
267.9297	0.3109	V	II	267.933
268.3058	0.1429	V	II	268.309
268.5665	0.0286	V	II	268.568
268.7945	1	V	II	268.796
269.0229	0.2316	V	II	269.025
269.0770	0	V	II	269.079
270.0916	0.5261	V	II	270.094
270.2165	0.2767	V	II	270.219
270.5197	0.0219	V	II	270.520
270.6133	0.4081	V	II	270.617
270.6680	0.1237	V	II	270.670
270.7846	0.0599	V	II	270.786
271.1706	0.0766	V	II	271.174
271.3037	0.0305	V	II	271.305
271.4191	0.0573	V	II	271.420
271.5648	0.2959	V	II	271.568
272.8619	0.1525	V	II	272.864
273.9687	0.0519	V	II	273.971
274.2404	0.0254	V	II	274.241
275.3383	0.0738	V	II	275.341
276.0677	0.0402	V	II	276.071
276.5648	0.0957	V	II	276.568
276.6441	0.0347	V	II	276.646
276.8546	0.0711	V	II	276.857
277.4250	0.0539	V	II	277.428
277.7743	0.0405	V	II	277.775
280.3449	0.0507	V	II	280.347
285.4327	0.0346	V	II	285.434
288.0016	0.0668	V	II	288.003
288.2478	0.0905	V	II	288.249
288.4763	0.0873	V	II	288.479
288.9607	0.0837	V	II	288.961

Table E-23. V ICP emission lines, part 1.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
289.1622	0.1717	V	II	289.164
289.2630	0.2305	V	II	289.265
289.3296	0.2489	V	II	289.331
289.6185	0.0540	V	II	289.620
290.3058	0.0491	V	II	290.307
290.6435	0.1673	V	II	290.645
290.7454	0.0741	V	II	290.746
290.8800	0.3329	V	II	290.881
291.0004	0.1485	V	II	291.001
291.0372	0.1041	V	II	291.038
291.1046	0.0991	V	II	291.105
292.0372	0.0599	V	II	292.038
292.3996	0.3808	V	II	292.402
293.0794	0.0951	V	II	293.080
293.4382	0.0183	V	II	293.439
294.1374	0.0540	V	II	294.137
294.4548	0.0915	V	II	294.457
295.0330	0.0192	V	II	295.034
295.2052	0.0564	V	II	295.207
295.7504	0.0231	V	II	295.752
296.8363	0.0299	V	II	296.837
297.6501	0.0180	V	II	297.652
309.3089	0.0347	V	II	309.311
310.2274	0.0250	V	II	310.230
311.0698	0.0180	V	II	311.071

Table E-24. V ICP emission lines, part 2.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
198.8740	0.0781	W	II	198.941
200.8085	0.1444	W	II	200.807
201.4229	0.0489	W	II	201.423
202.6076	0.1627	W	II	202.608
202.9984	0.2103	W	II	202.998
203.5883	0.2196	W	II	203.589
204.3537	0.0517	W	II	204.352
204.9627	0.1254	W	II	204.963
205.4673	0.1980	W	II	205.468
206.5561	0.1028	W	II	206.557
206.7518	0.0536	W	II	206.752
207.1194	0.1394	W	II	207.121
207.5578	0.0585	W	II	207.559
207.9106	0.7154	W	II	207.911
208.8192	0.3147	W	II	208.819
208.9149	0.1522	W	II	208.914
209.4742	0.4808	W	II	209.475
209.8590	0.2710	W	II	209.860
210.0662	0.1695	W	II	210.067
210.5750	0.0376	W		NLV
210.6178	0.1844	W	II	210.618
211.0325	0.1100	W	II	211.034
211.6939	0.0281	W	II	211.694
211.8864	0.2420	W	II	211.887
213.8145	0.1438	W	II	213.815
215.3544	0.2110	W		215.355
216.6311	0.4526	W	II	216.632
218.9493	0.3245	W	II	218.949
219.4521	0.1543	W	II	219.452
220.4471	0.7573	W	II	220.448
220.6594	0.1061	W	II	220.659
220.7980	0.0754	W	II	220.799
221.6003	0.2817	W	II	221.601
222.5886	0.5540	W	II	222.589
222.9620	0.5013	W	II	222.963
224.1077	0.1304	W	II	224.108
224.5190	0.1733	W	II	224.520
224.6637	0.1135	W	II	224.663
224.8269	0.1891	W	II	224.826
224.8745	1	W	II	224.875
224.9879	0.0303	W		224.984
225.0722	0.2807	W	II	225.073
225.1125	0.1451	W	II	225.114
226.3509	0.1030	W	II	226.353
226.6119	0.2011	W	II	226.612
227.0225	0.3413	W	II	227.024
229.4529	0.1913	W	II	229.454
230.3808	0.4578	W	II	230.383
231.5008	0.2746	W	II	231.502
232.3022	0.0970	W	II	232.303
232.6084	0.5749	W	II	232.609
232.8304	0.1142	W	II	232.831
233.3767	0.1675	W	II	233.377
233.9151	0.0474	W	II	233.916
235.8801	0.0998	W	II	235.881

Table E-25. W ICP emission lines, part 1.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
236.4206	0.0789	W	II	236.422
237.0029	0.0631	W	II	237.004
237.4453	0.0405	W	II	237.445
239.0355	0.1362	W	II	239.037
239.2920	0.1990	W		239.293
239.5740	0.0480	W	II	239.571
239.7097	0.6523	W	II	239.709
239.7970	0.0258	W	I	239.798
240.5565	0.0282	W	I	240.558
242.7473	0.1204	W	II	242.749
243.5942	0.0415	W	I	243.596
244.6374	0.0872	W	II	244.639
245.1464	0.0578	W		245.148
246.6510	0.0724	W	II	246.652
247.7783	0.0423	W	II	247.780
248.9204	0.0465	W	II	248.923
252.2036	0.0451	W	II	252.204
255.1329	0.0414	W	I	255.135
255.5085	0.1566	W	II	255.509
256.3139	0.0535	W	II	256.316
256.9284	0.0321	W		256.925
257.1434	0.3169	W	II	257.145
257.2343	0.0277	W	II	257.235
257.9238	0.0332	W	II	257.926
257.9527	0.0724	W	II	257.954
258.1181	0.0468	W	II	258.120
258.9141	0.1614	W	II	258.917
259.8729	0.0565	W	II	259.874
260.3002	0.0776	W	II	260.302
261.5420	0.0602	W	II	261.545
265.8022	0.4520	W	II	265.804
266.4322	0.1068	W	II	266.432
269.7701	0.3573	W	II	269.771
276.4247	0.4129	W	II	276.427

Table E-26. W ICP emission lines, part 2.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
229.1106	0.0350	Zr	II	229.115
229.4034	0.0821	Zr	II	229.408
229.5481	0.0329	Zr	II	229.548
231.7246	0.0355	Zr	II	231.727
232.4754	0.0239	Zr	II	232.477
233.0332	0.0246	Zr	II	233.038
235.7406	0.0262	Zr	II	235.745
237.2945	0	Zr	II	237.293
238.7204	0	Zr	II	238.717
244.9828	0.0400	Zr	II	244.983
245.7419	0.0531	Zr	II	245.743
253.2462	0.0285	Zr	II	253.247
254.2102	0.0506	Zr	II	254.210
255.0732	0.0548	Zr	II	255.071
256.8854	0.5032	Zr	II	256.885
257.145	0.4446	Zr	II	257.142
258.3383	0.0261	Zr	II	258.338
258.906	0.0439	Zr	II	258.902
263.0892	0.1178	Zr	II	263.091
263.9061	0.0914	Zr	II	263.908
264.3384	0.0204	Zr	II	264.339
265.0361	0.0403	Zr	II	265.038
266.7784	0.0814	Zr	II	266.779
267.0963	0.0236	Zr	II	267.096
267.8622	1	Zr	II	267.863
268.9444	0.0262	Zr	II	268.945
269.2587	0.0301	Zr	II	269.260
269.3514	0.0833	Zr	II	269.352
269.4056	0	Zr	II	269.406
269.9597	0.0512	Zr	II	269.960
270.0107	0.5261	Zr	II	270.013
271.1493	0.1277	Zr	II	271.151
271.238	0	Zr	II	271.242
271.4244	0.0851	Zr	II	271.425
272.2601	0.7193	Zr	II	272.261
272.6467	0.4484	Zr	II	272.649
273.2701	0.2650	Zr	II	273.272
273.4827	0.6716	Zr	II	273.484
274.0483	0.0662	Zr	II	274.050
274.1534	0.0816	Zr	II	274.155
274.2531	0.6318	Zr	II	274.256
274.5836	0.4259	Zr	II	274.586
275.2177	0.4551	Zr	II	275.221
275.8786	0.2395	Zr	II	275.881
276.8713	0.0924	Zr	II	276.873
277.4139	0.0820	Zr	II	277.415
279.6888	0.0675	Zr	II	279.690
281.0897	0.0827	Zr	II	281.091
281.8710	0.1112	Zr	II	281.874
282.5547	0.1118	Zr	II	282.556
284.4558	0.2017	Zr	II	284.458
285.1952	0.0856	Zr	II	285.197
286.9792	0.0733	Zr	II	286.981
291.5978	0.0448	Zr	II	291.599
292.6968	0.0513	Zr	II	292.699

Table E-27. Zr ICP emission lines.

Wavelength (nm)	Rel. Intensity	Element	State	Literature wavelength (nm)
193.0278	0	C	I	193.091
247.8568	0	C	I	247.857
214.4391	1	Cd	II	214.438
226.5005	0.9168	Cd	II	226.502
228.8010	0.8002	Cd	I	228.802
194.0932	0.0073	Hg		NLV
194.1628	0.6966	Hg	II	194.227
194.2984	0.0043	Hg		NLV
208.8897	0	Hg		NLV
208.9581	0.1461	Hg		NLV
249.6772	0	Hg		NLV
249.7719	0	Hg		NLV
253.6520	1	Hg	I	253.652
277.9829	0.003	Mg	I	277.983
279.0772	0.006	Mg	II	279.079
279.5531	1	Mg	II	279.553
279.8001	0.01	Mg	II	279.806
280.2710	0.6	Mg	II	280.270
285.2129	0.03	Mg	I	285.213
205.8150	0.0462	Si	I	205.813
212.4117	1	Si	I	212.412
221.0888	0.4834	Si	I	221.089
221.1736	0.1515	Si	I	221.174
221.6666	0.8729	Si	I	221.667
221.8053	0.1582	Si	I	221.806
243.5141	0.1504	Si	I	243.515
251.4315	0.1371	Si	I	251.432
251.6100	0.5678	Si	I	251.611
251.9202	0	Si	I	251.921
252.4109	0	Si	I	252.412
252.8510	0.3171	Si	I	252.851
288.1576	0.6726	Si	I	288.158
202.5473	0.2768	Zn	II	202.551
206.1988	0.3010	Zn	II	206.191
213.8552	1	Zn	I	213.856

Table E-28. Miscellaneous ICP emission lines.

Appendix F

ICP spectra of 19 elements.

This appendix contains the ultraviolet ICP spectra of Cd, Co, Cr, Cu, Fe, La, Mg, Mn, Mo, Ni, Pb, Sn, Ta, Th, U, V, W, Zn, and Zr. Each spectrum shows the complete range from 185 to 315 nm compressed to fit onto a single page. The spectra were calculated from 64K interferograms sampled with a 4× clock. In conjunction with a solar blind detector this results in medium resolution (2 cm^{-1}) de-aliased spectra.

The horizontal axis is actually in units of wavenumbers but has been converted into nanometers for convenience, this results in a non-linear wavelength scale. The vertical axes of some of the spectra have been expanded to show minor lines in the baseline.

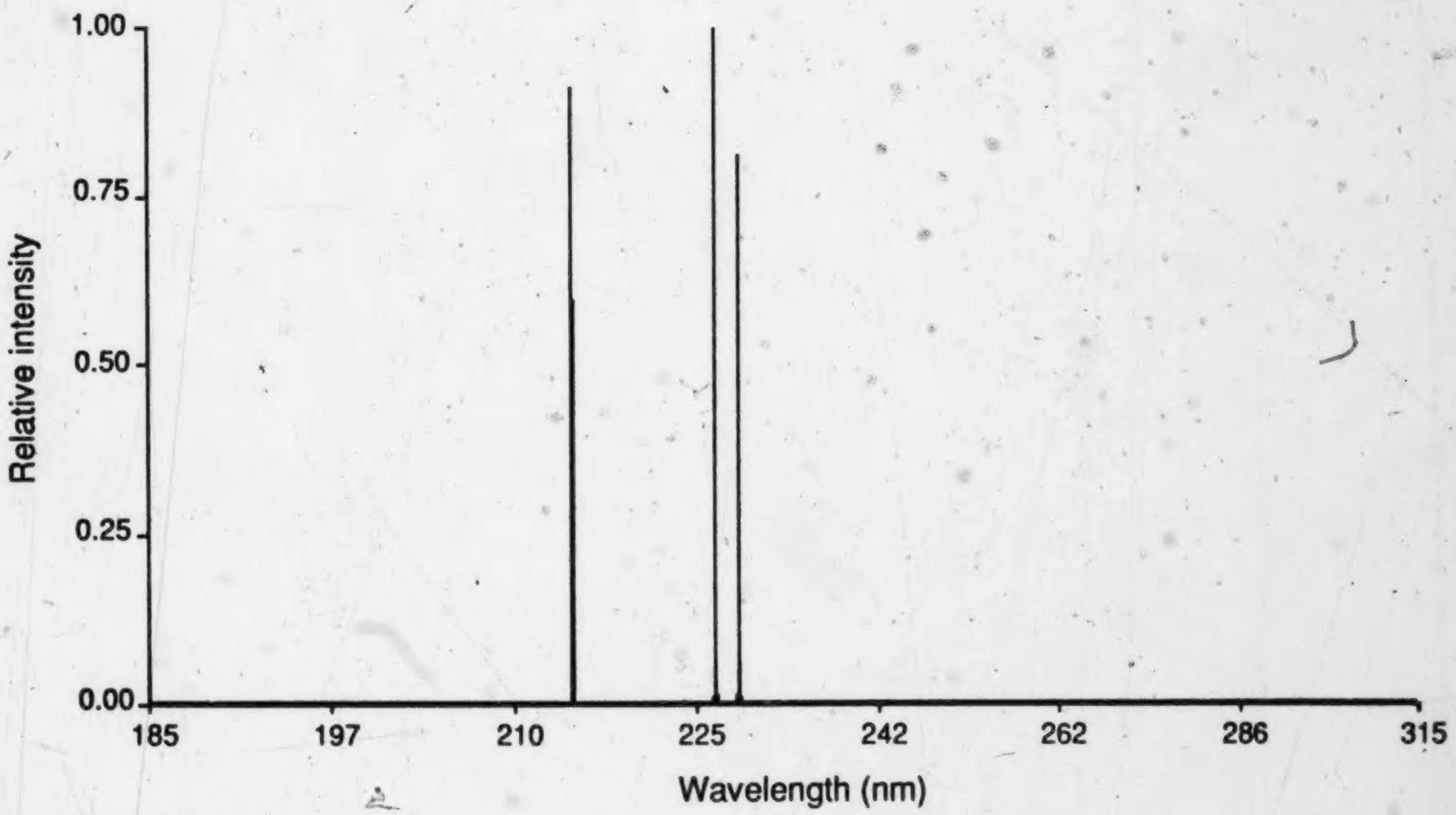


Figure F-1. The ICP ultraviolet spectrum of Cd.

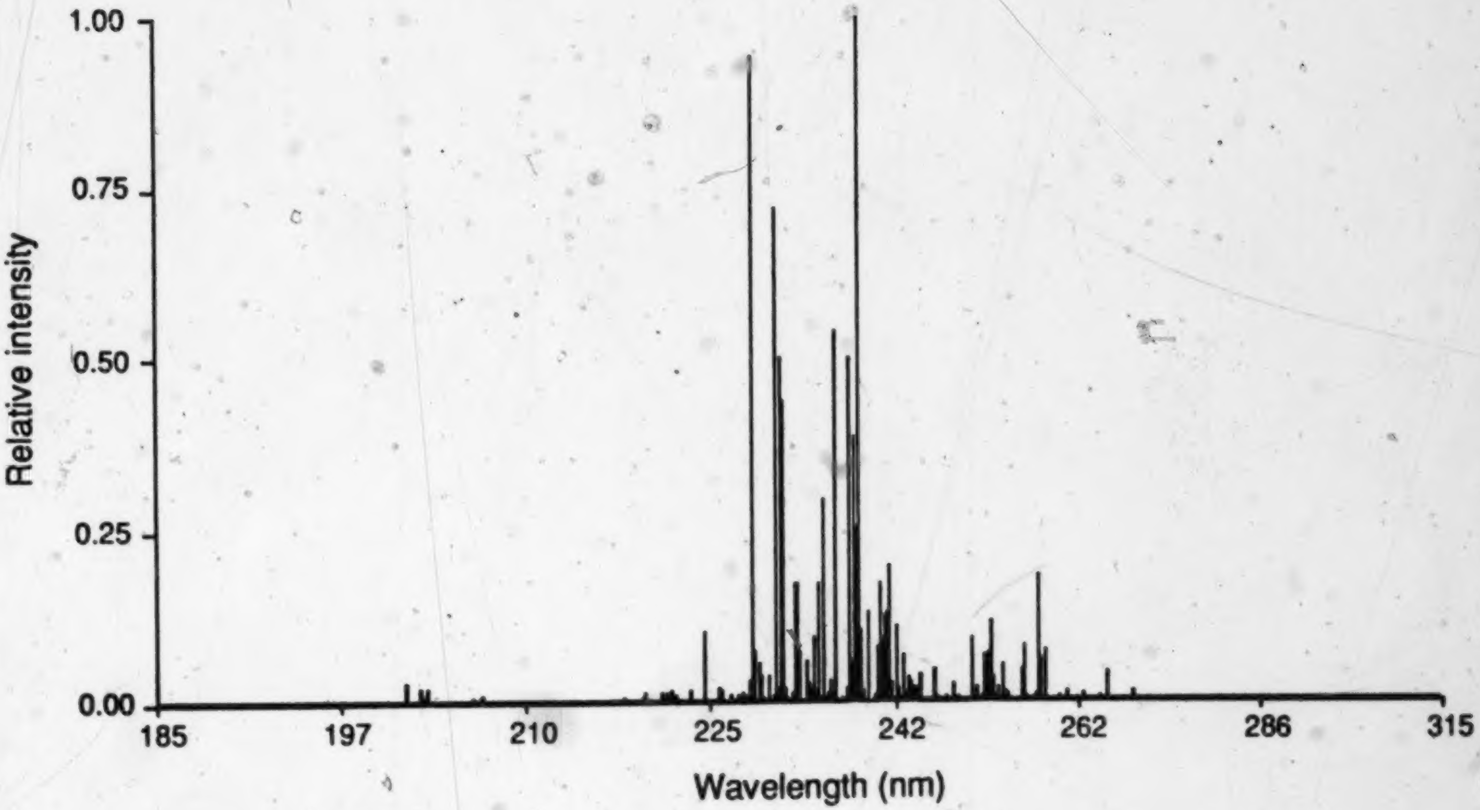


Figure F-2. The ICP ultraviolet spectrum of Co.

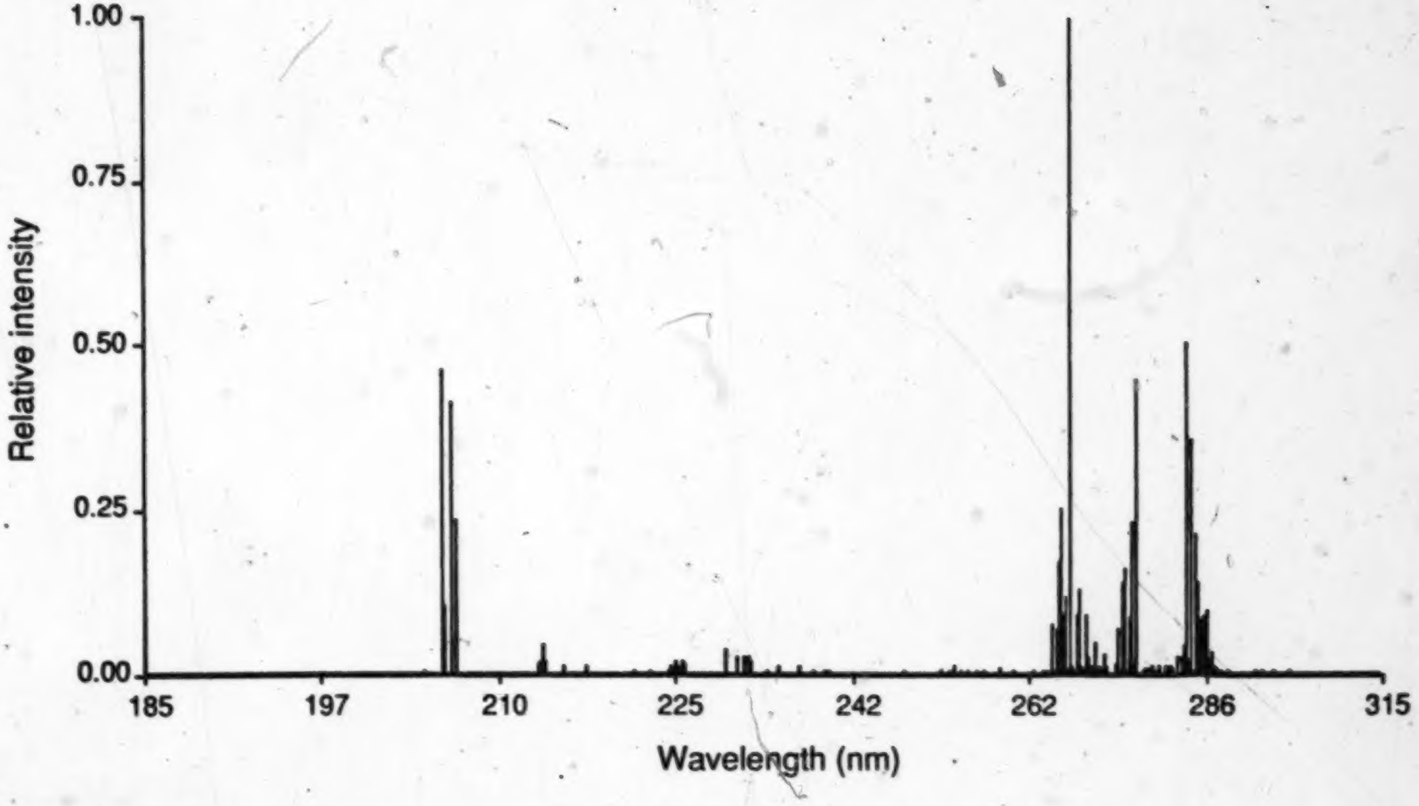


Figure F-3. The ICP ultraviolet spectrum of Cr.

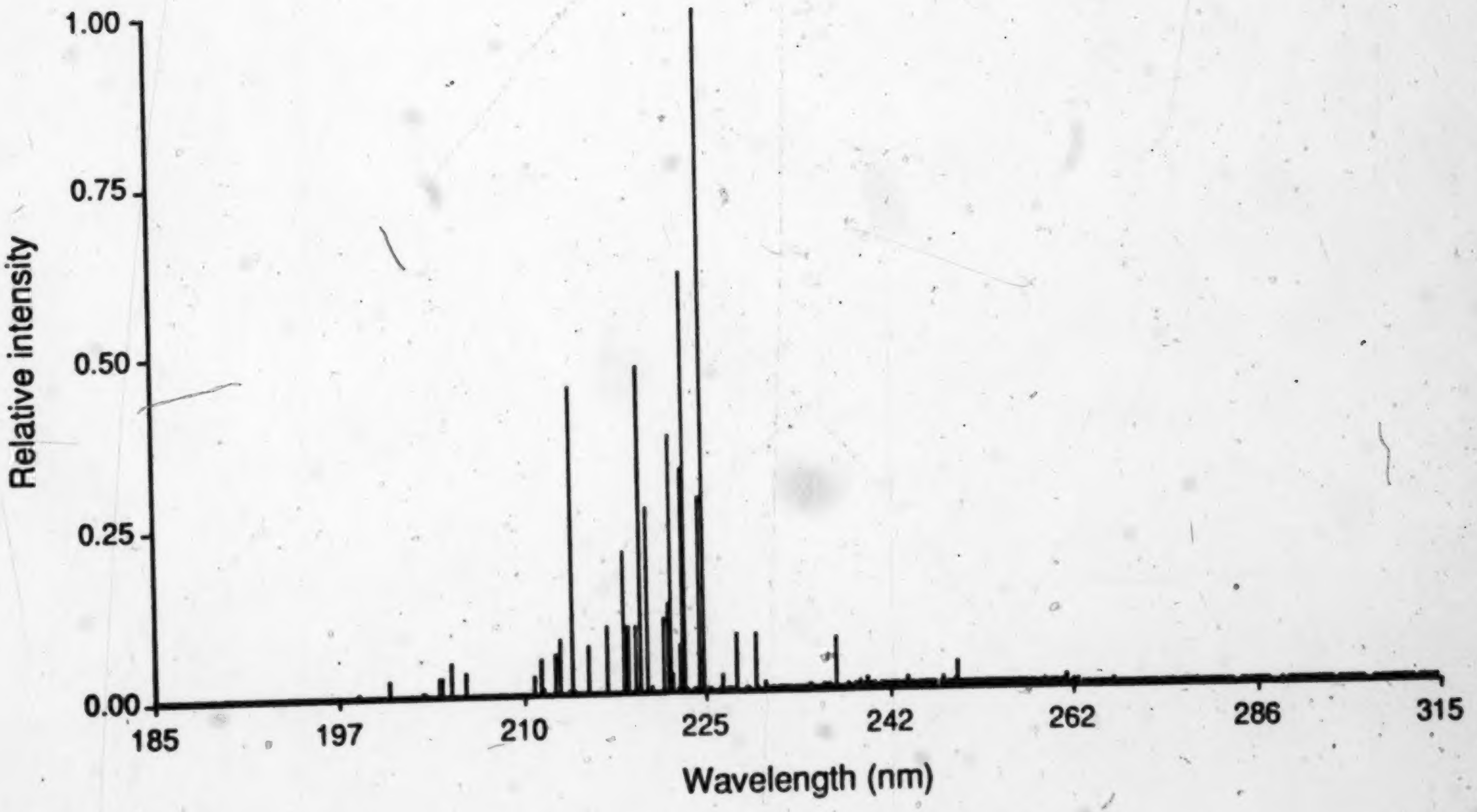


Figure F-4. The ICP ultraviolet spectrum of Cu.

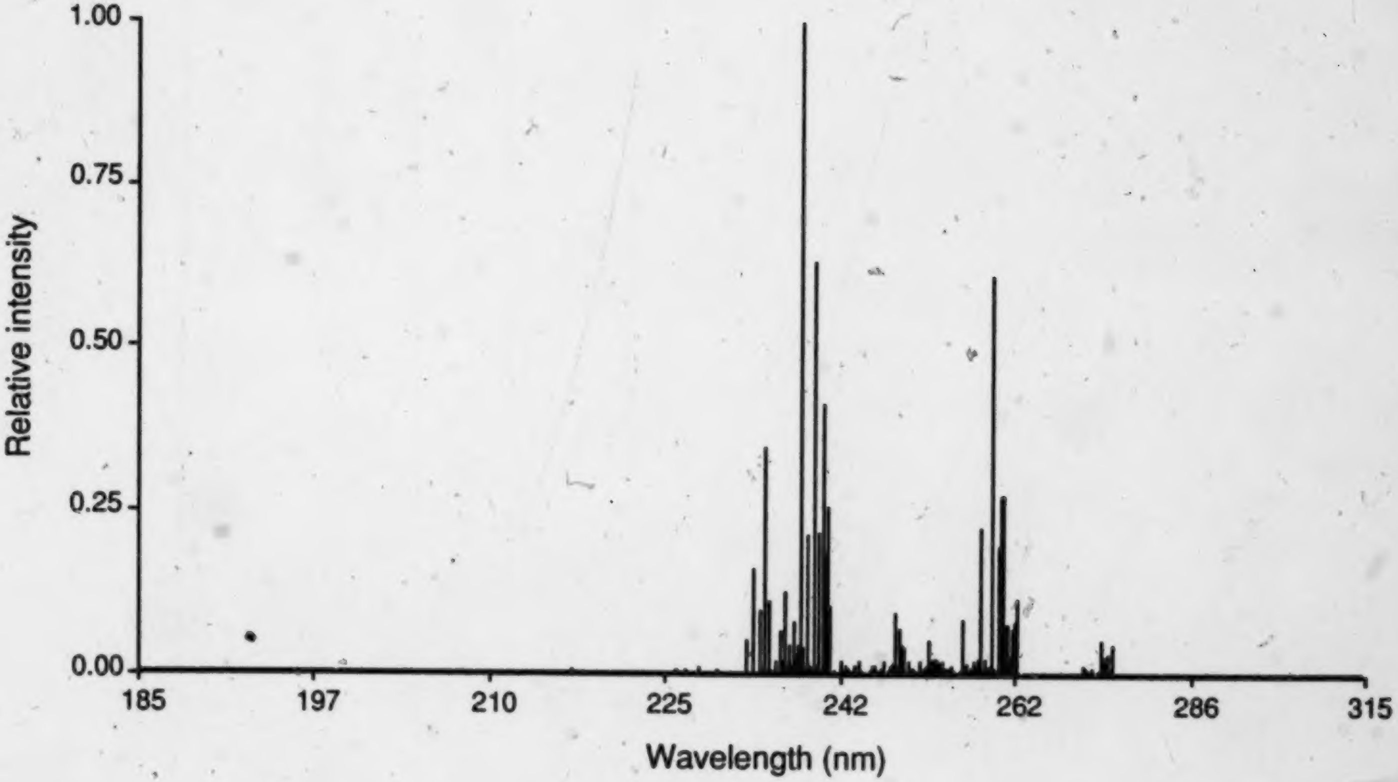


Figure F-5. The ICP ultraviolet spectrum of Fe.

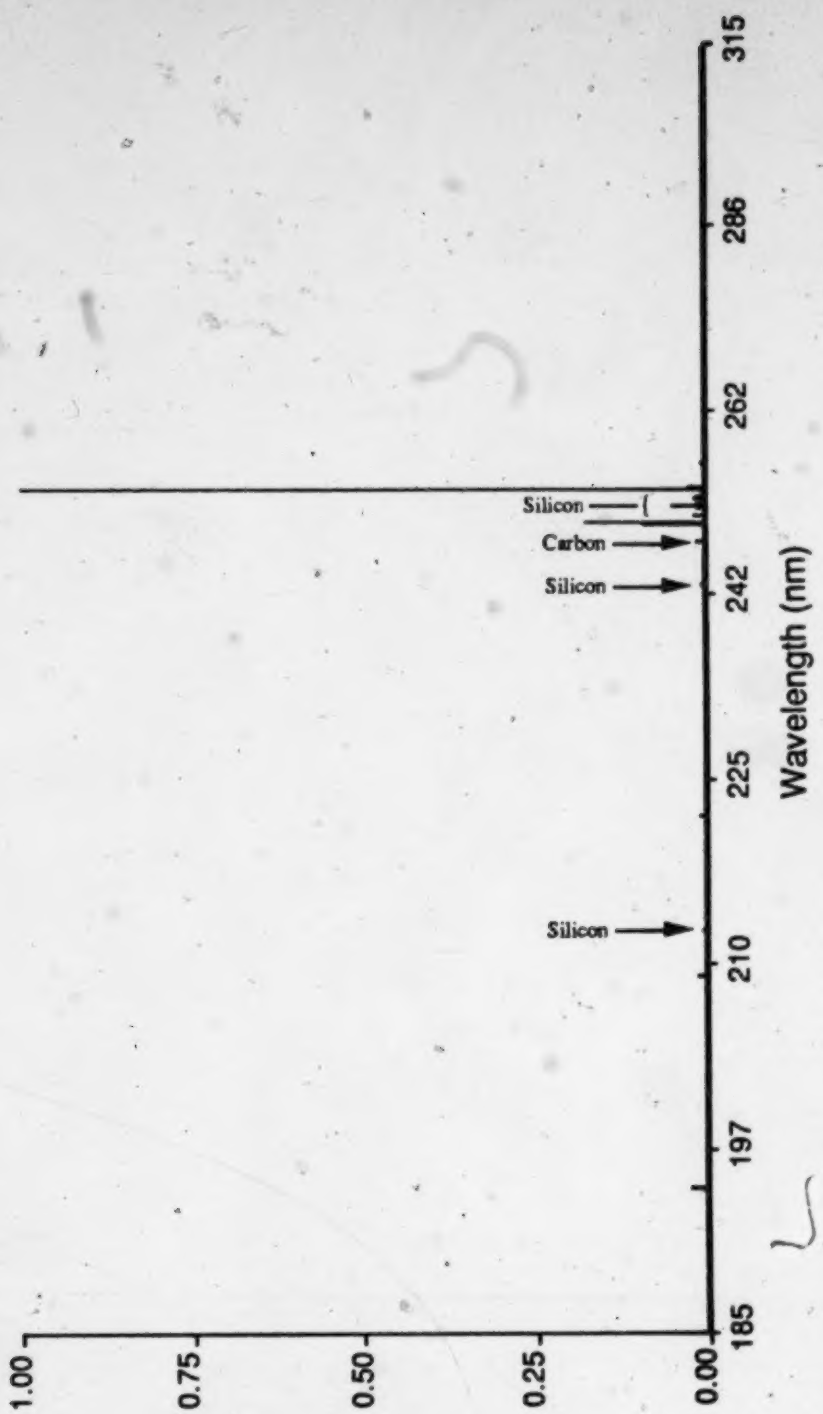


Figure F-6. The ICP ultraviolet spectrum of Hg.

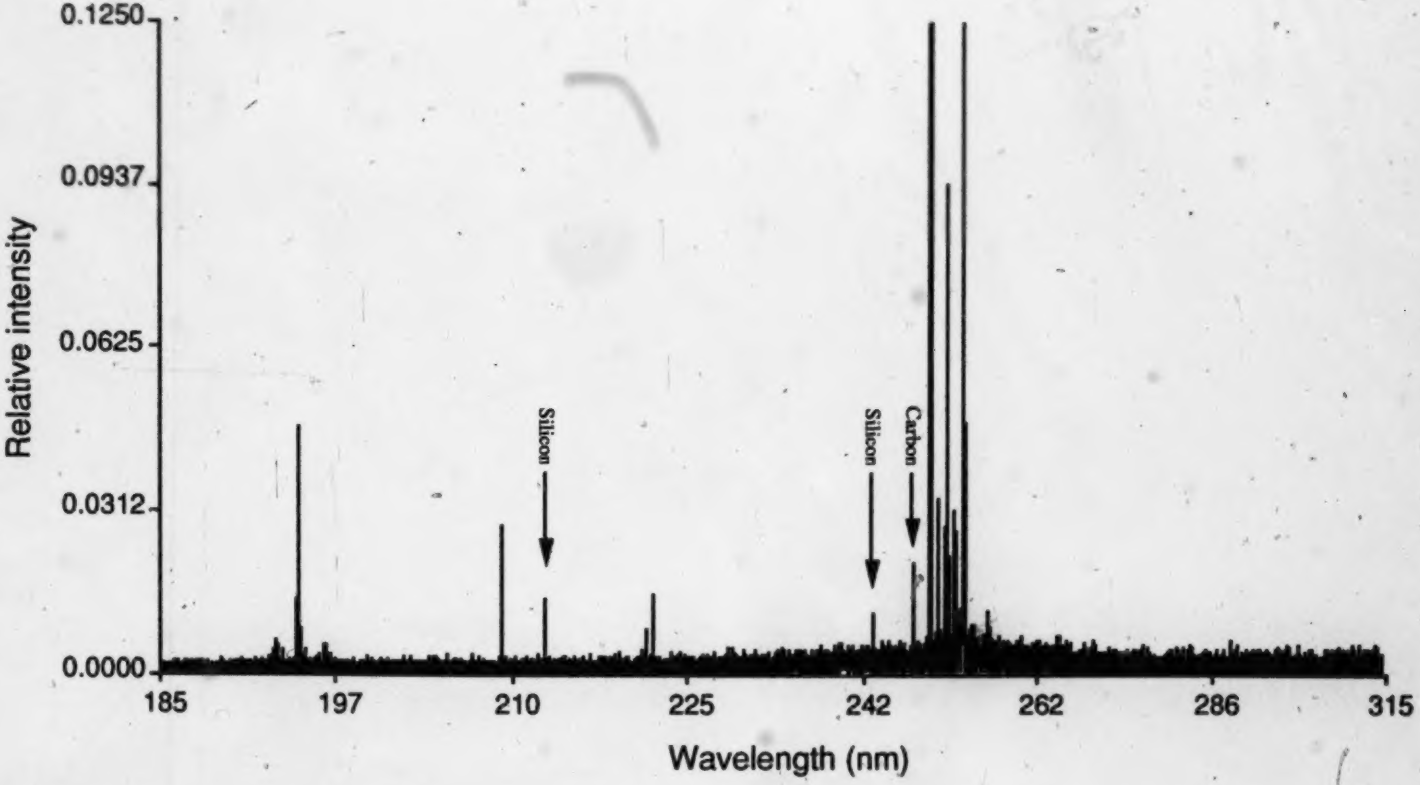


Figure F-7. Scale expanded Hg ICP spectrum showing minor spectral features.

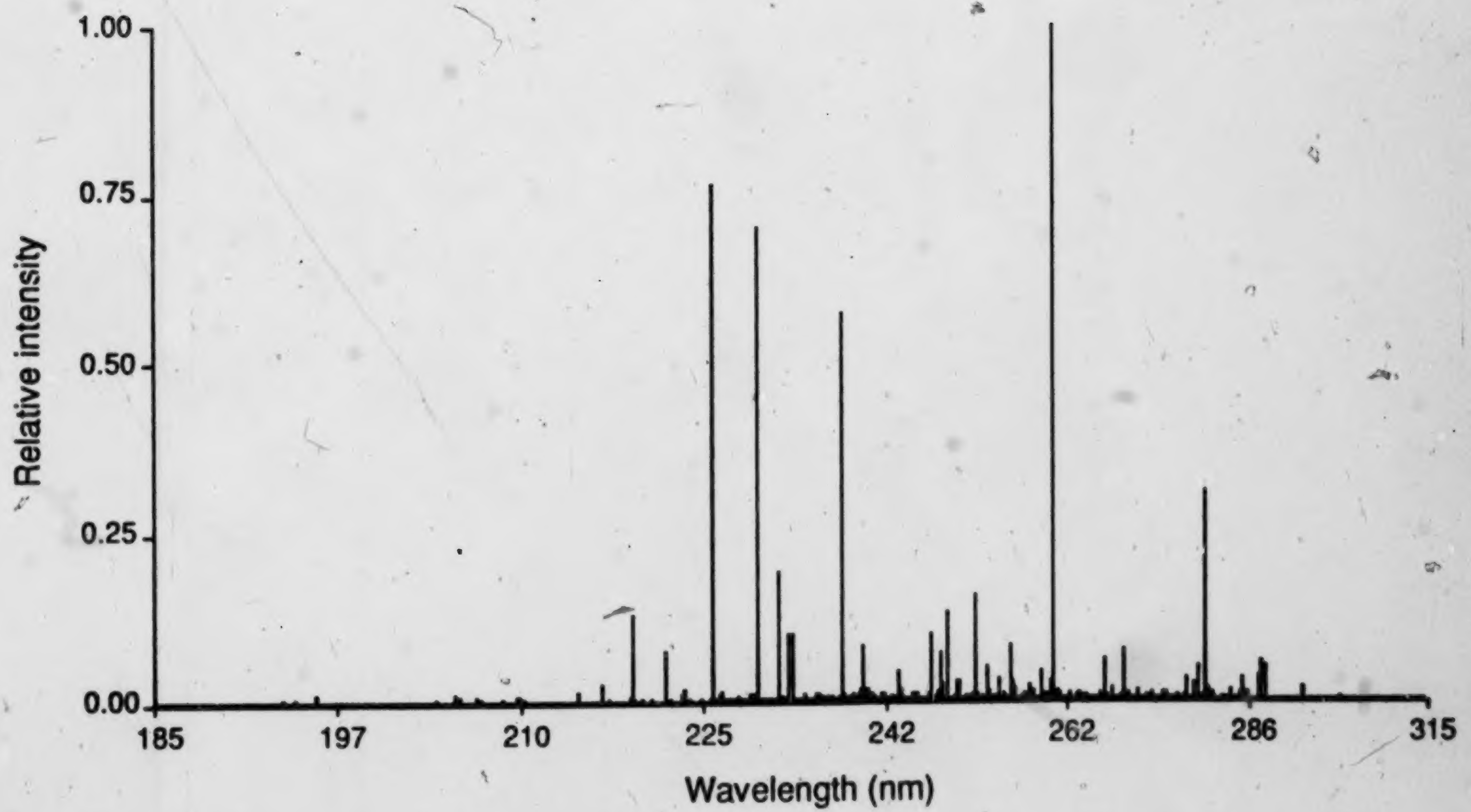


Figure F-8. The ICP ultraviolet spectrum of La.

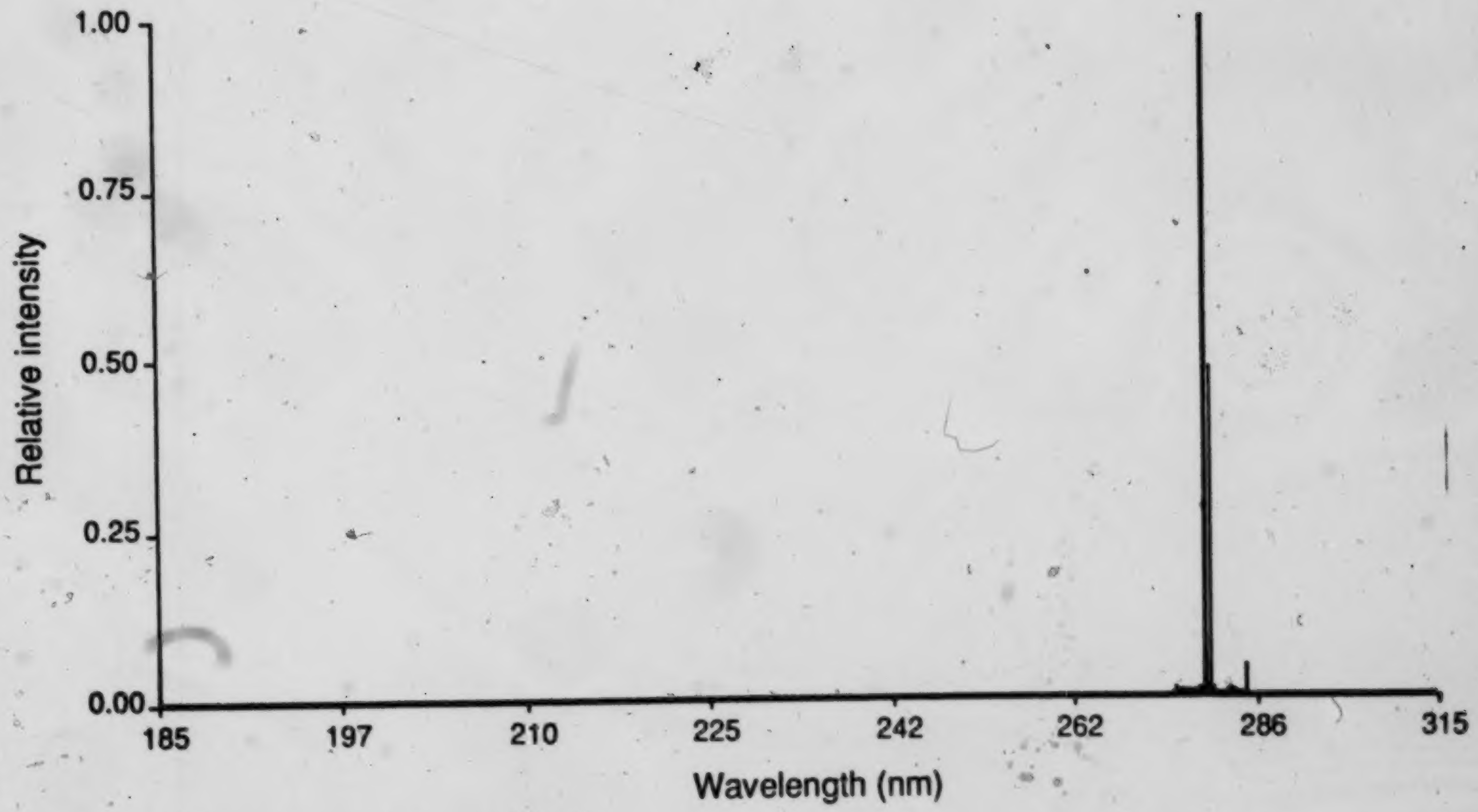


Figure F-9. The ICP ultraviolet spectrum of Mg.

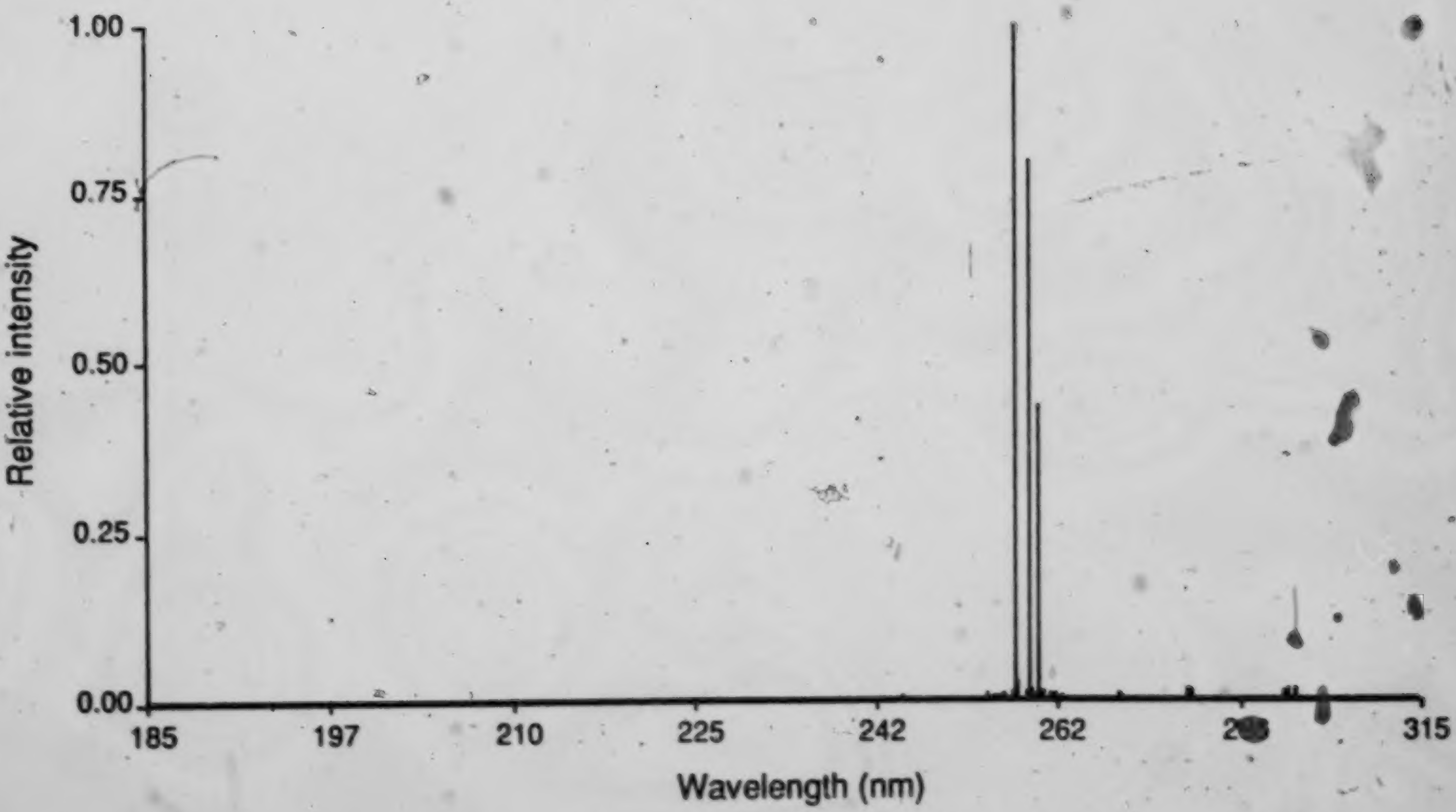


Figure F-10. The ICP ultraviolet spectrum of Mn.

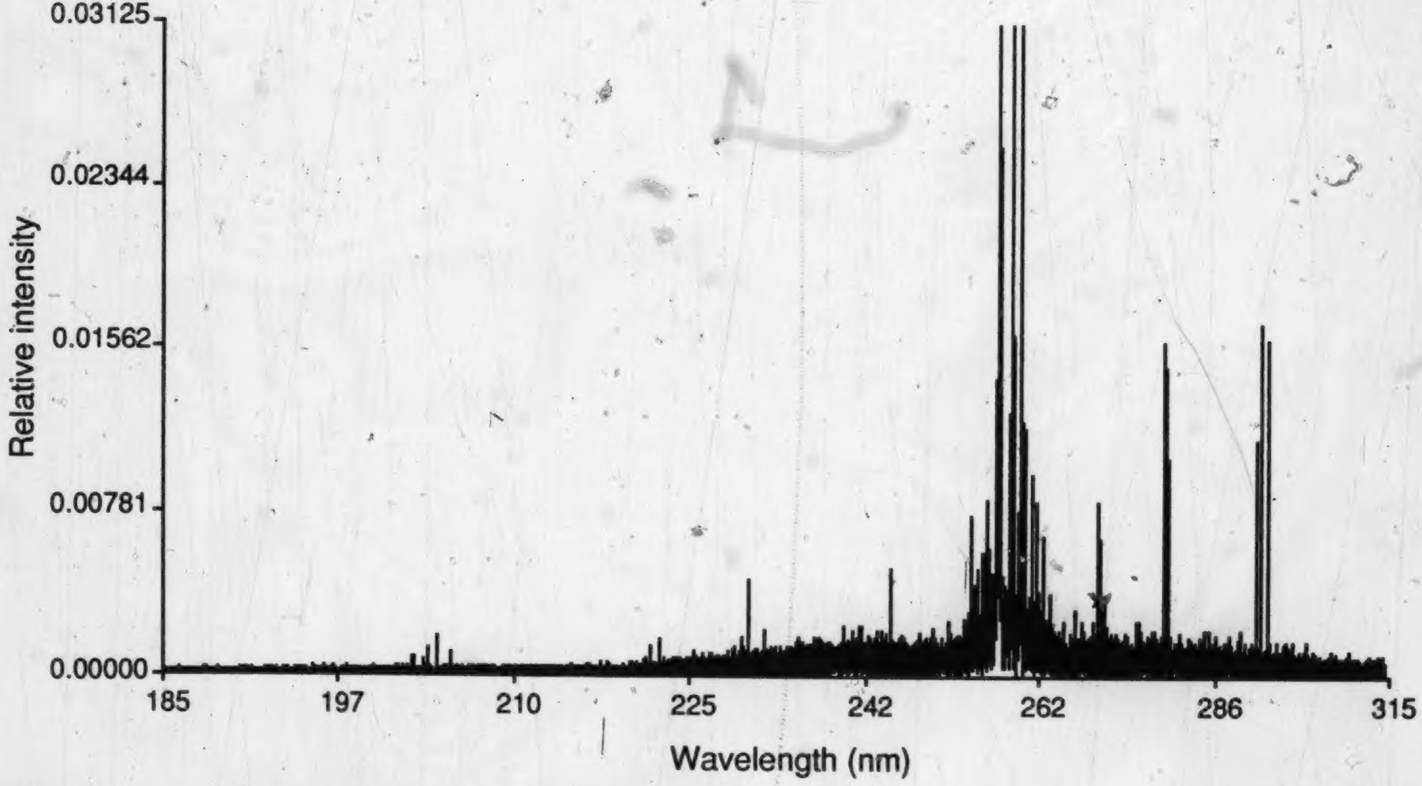


Figure F-11. Scale expanded Mn ICP spectrum showing minor spectral features.

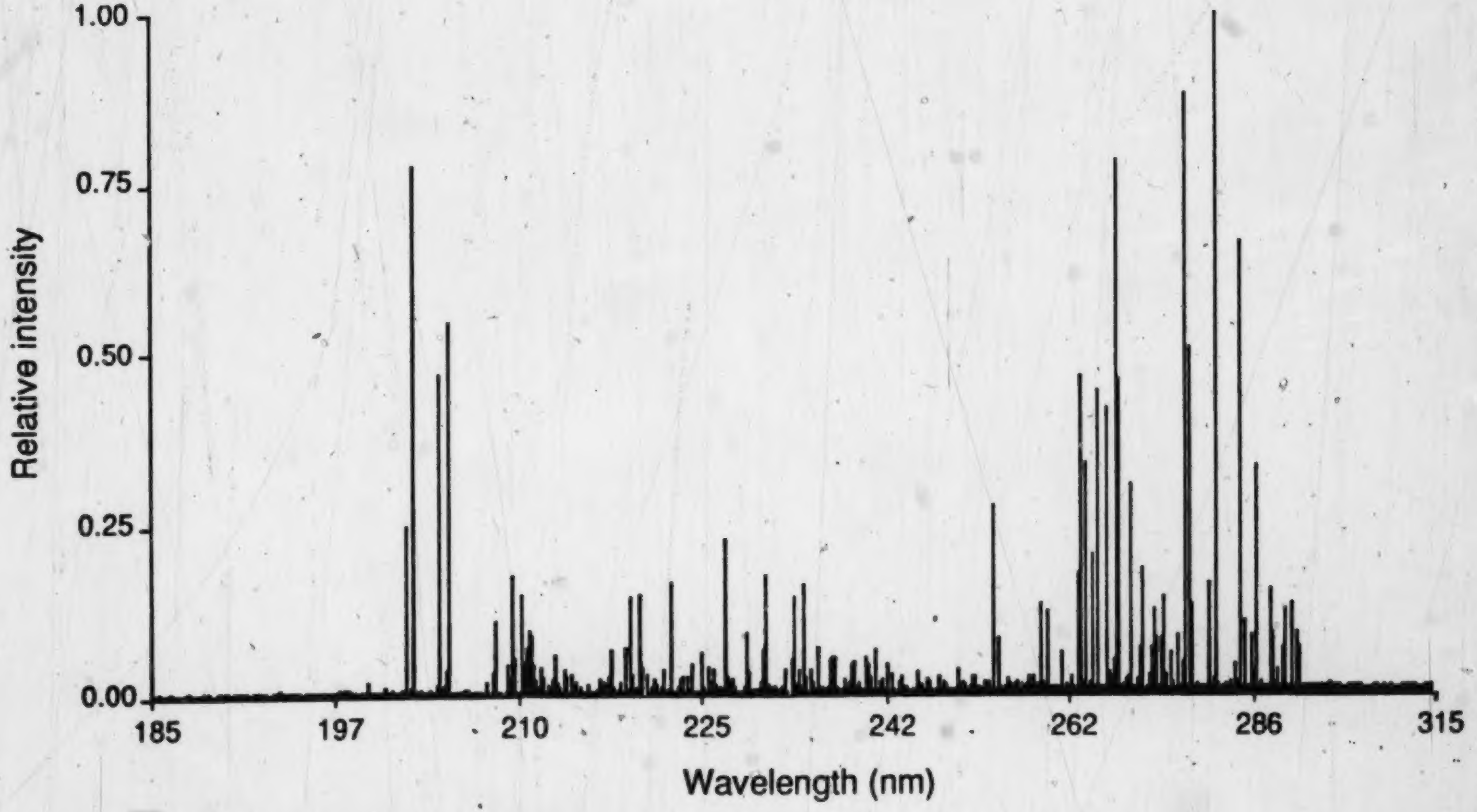


Figure F-12. The ICP ultraviolet spectrum of Mo.

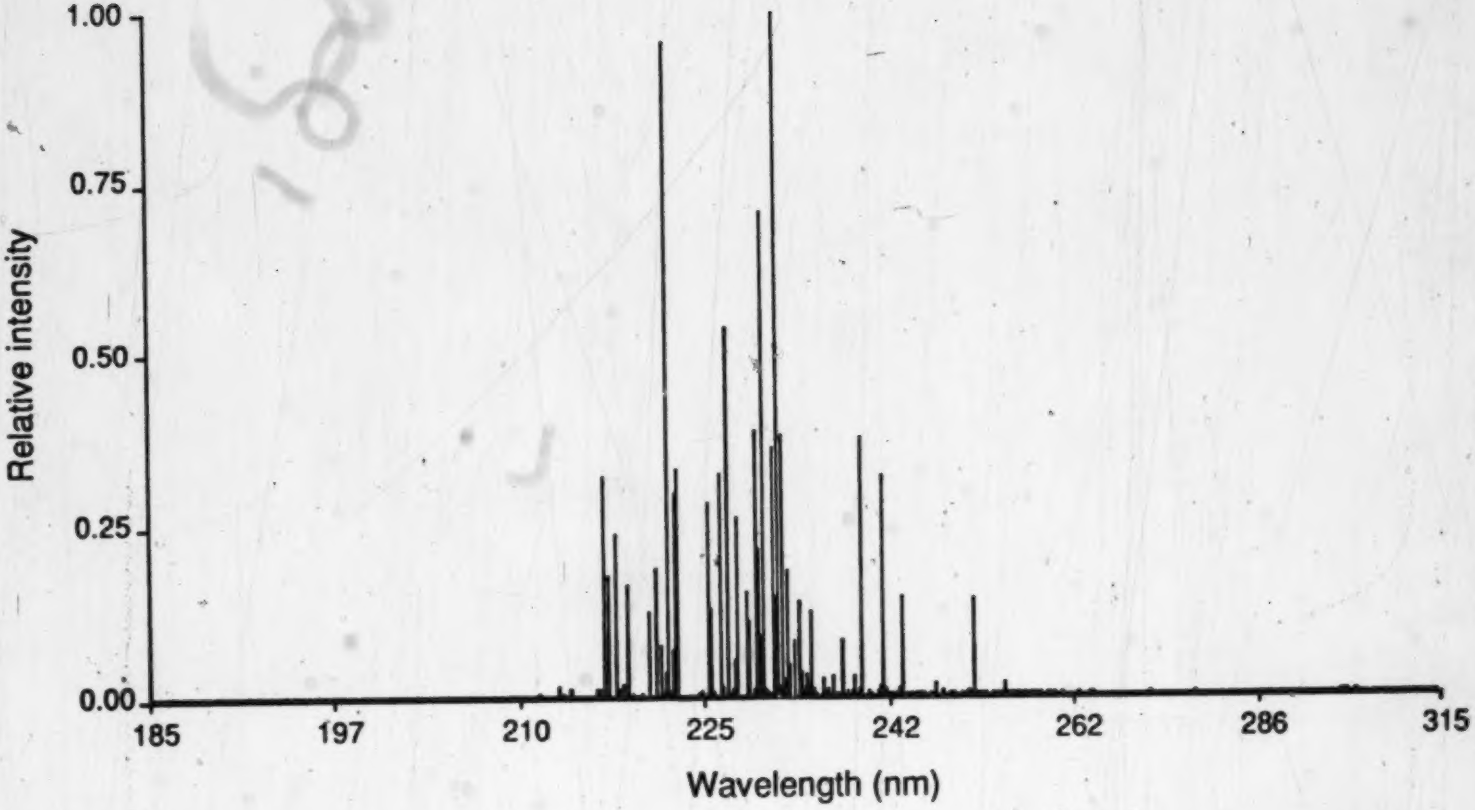


Figure F-13. The ICP ultraviolet spectrum of Ni.

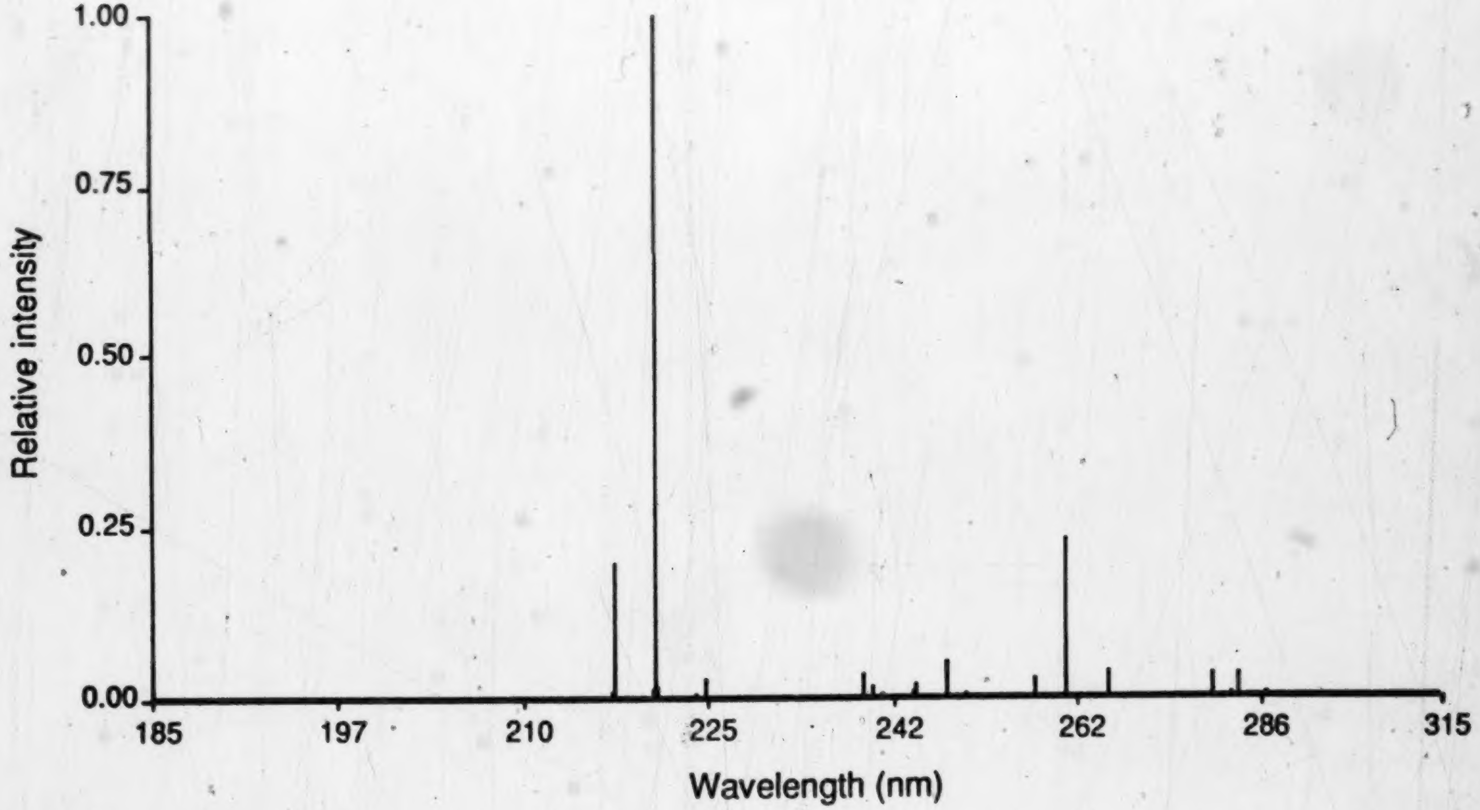


Figure F-14. The ICP ultraviolet spectrum of Pb.

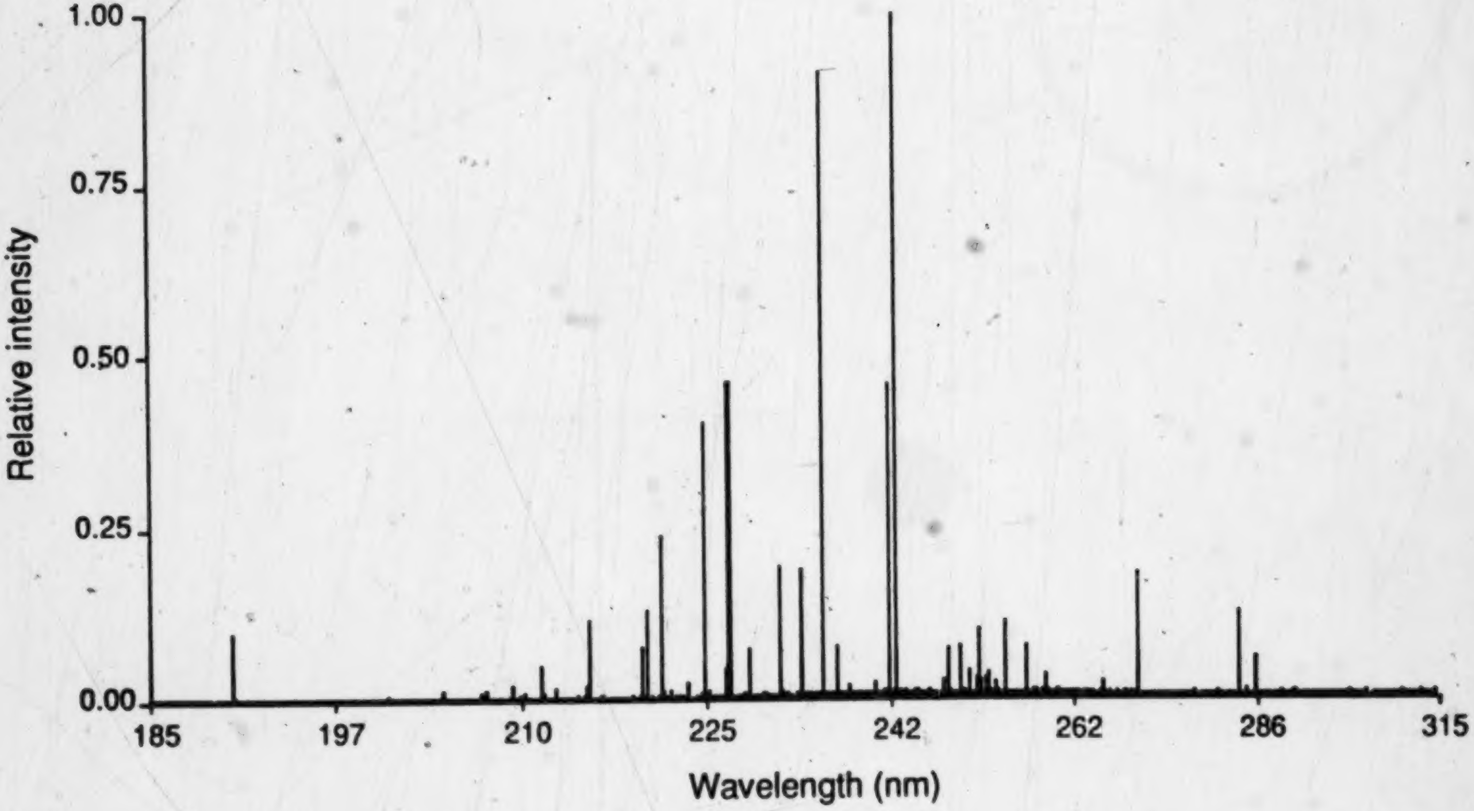


Figure F-15. The ICP ultraviolet spectrum of Sn.

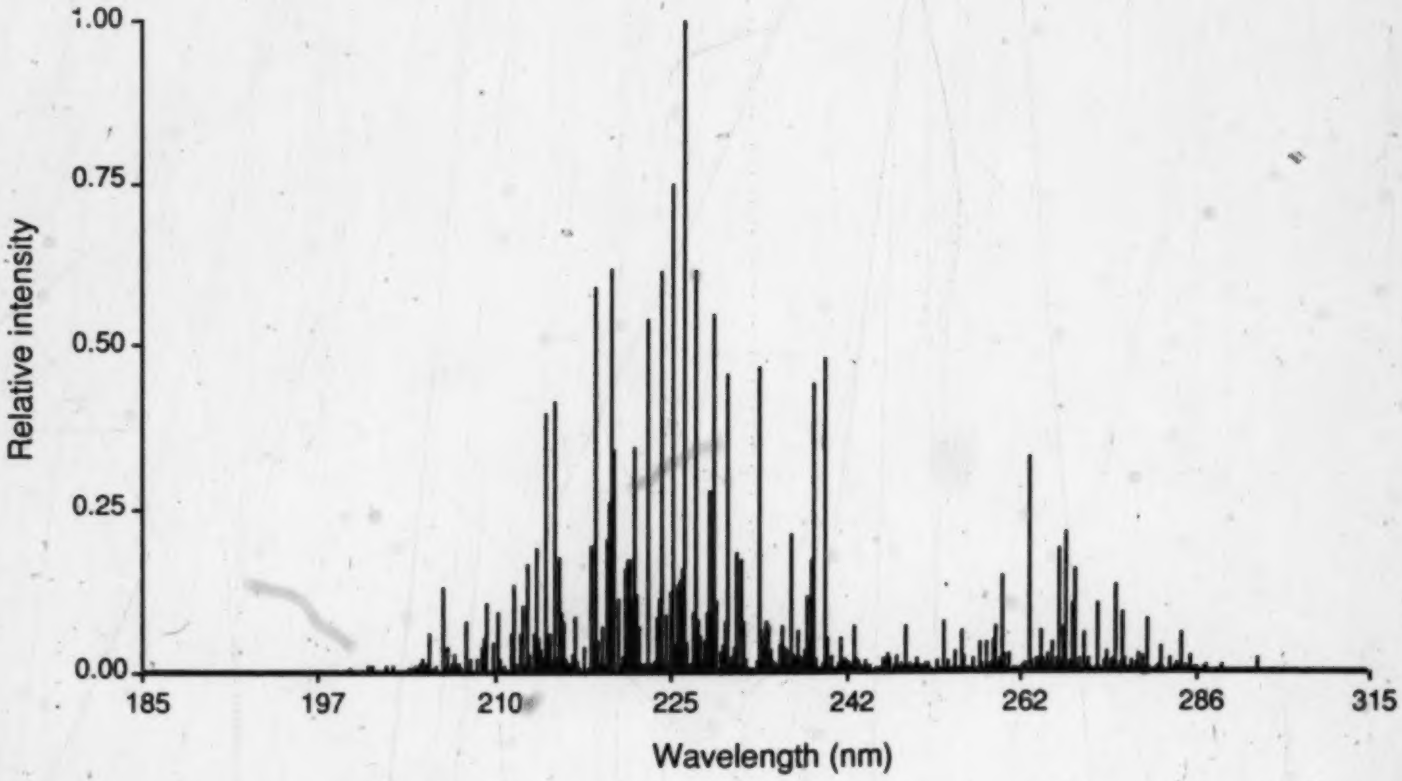


Figure F-16. The ICP ultraviolet spectrum of Ta.

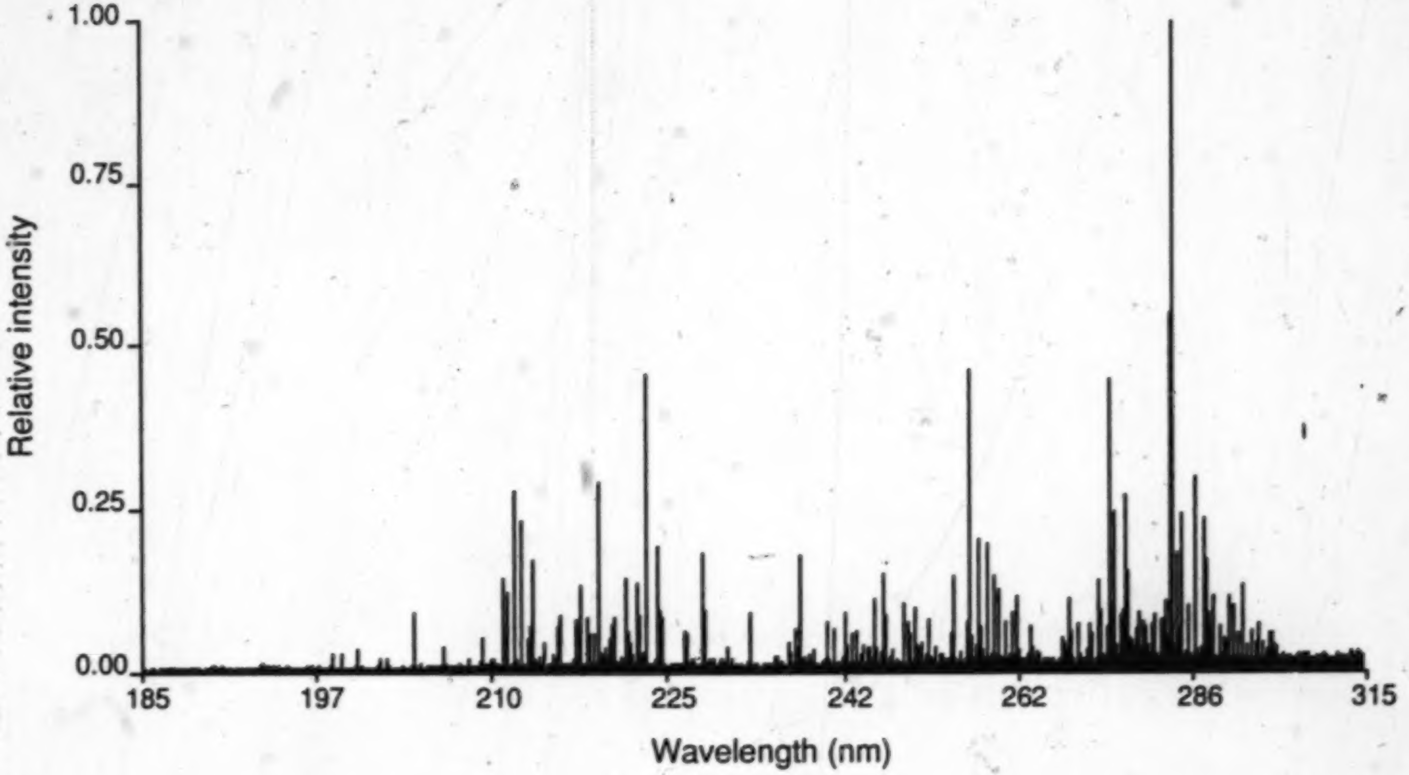


Figure F-17. The ICP ultraviolet spectrum of Th.

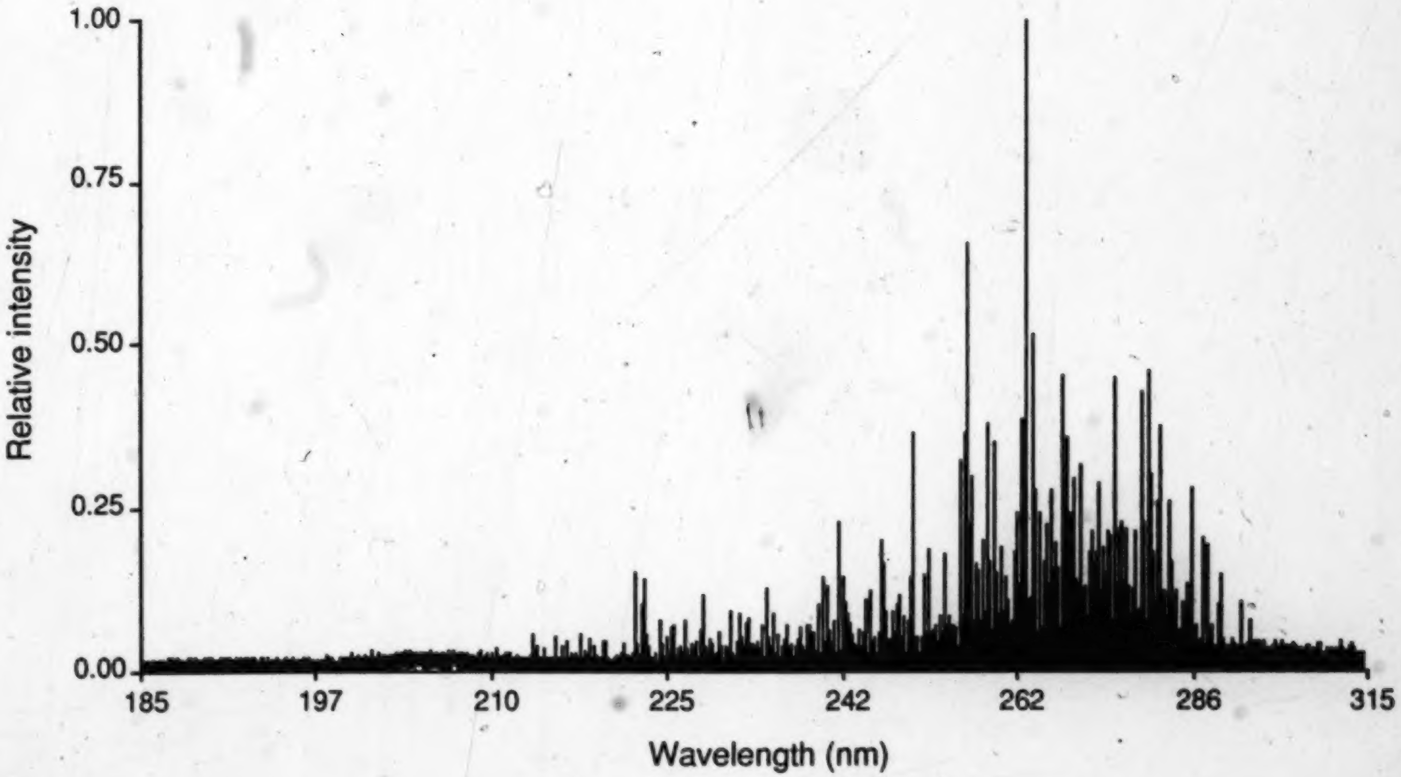


Figure F-18. The ICP ultraviolet spectrum of U.

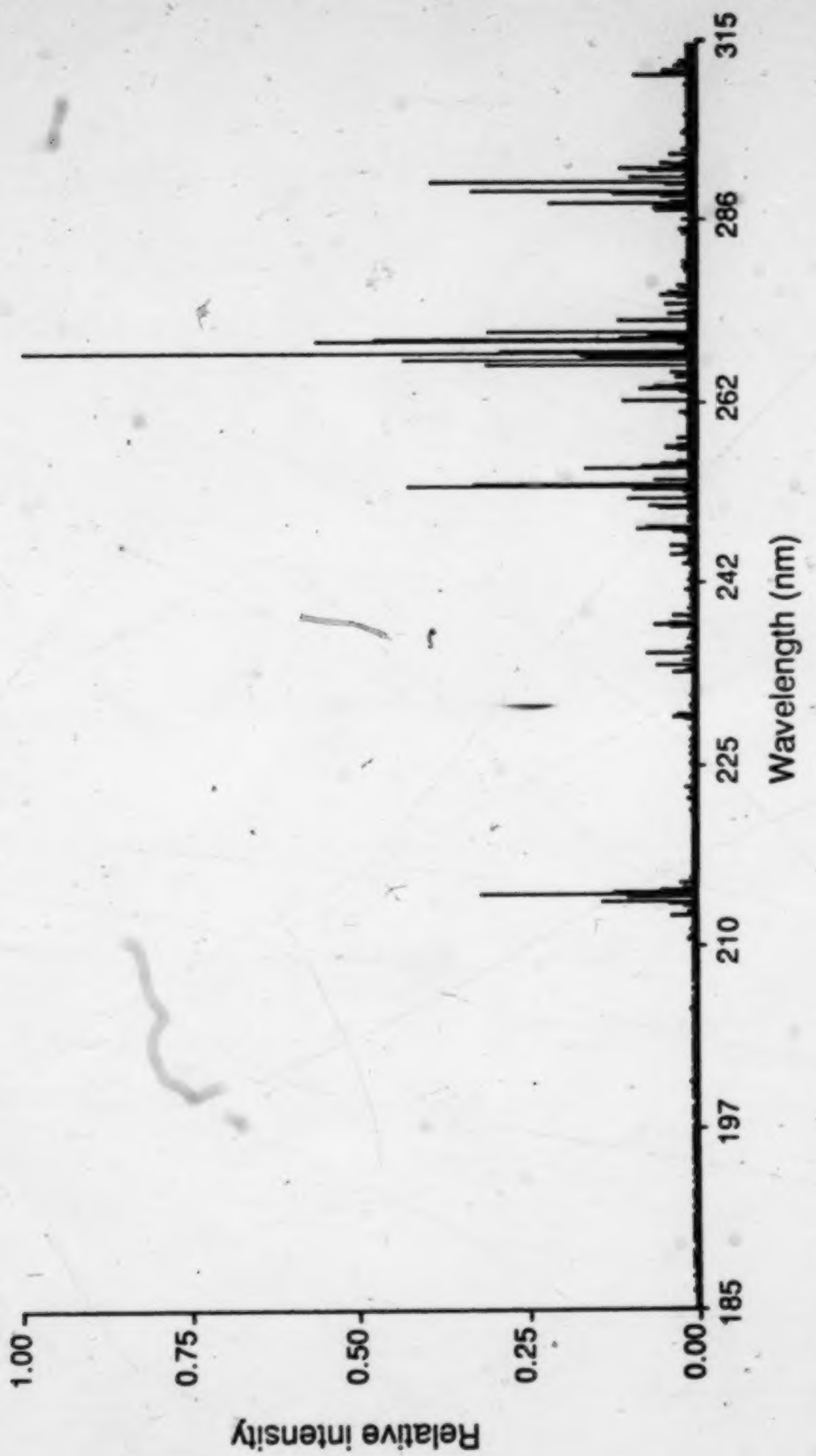


Figure F-19. The ICP ultraviolet spectrum of V.

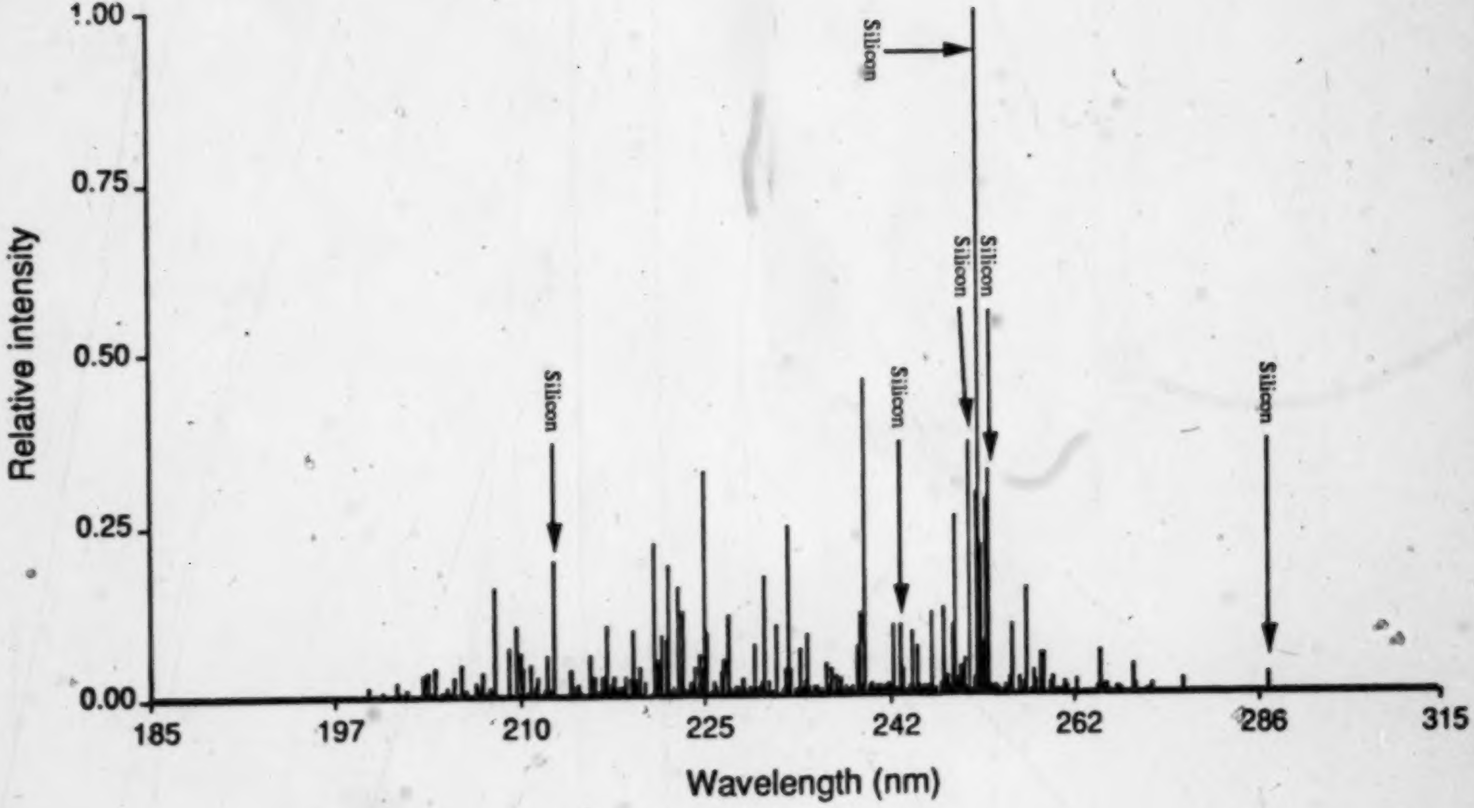


Figure F-20. The ICP ultraviolet spectrum of W.

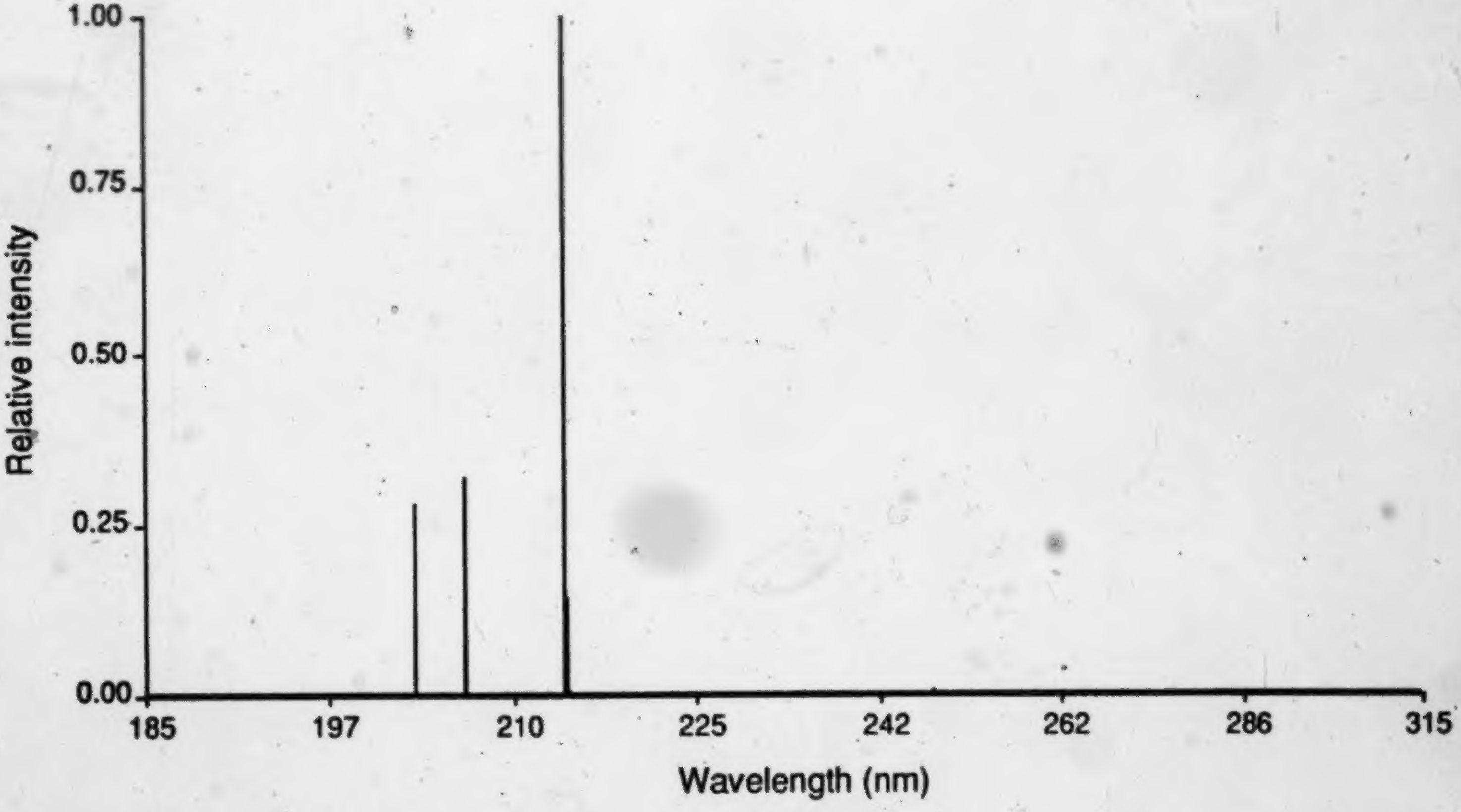


Figure F-21. The ICP ultraviolet spectrum of Zn.

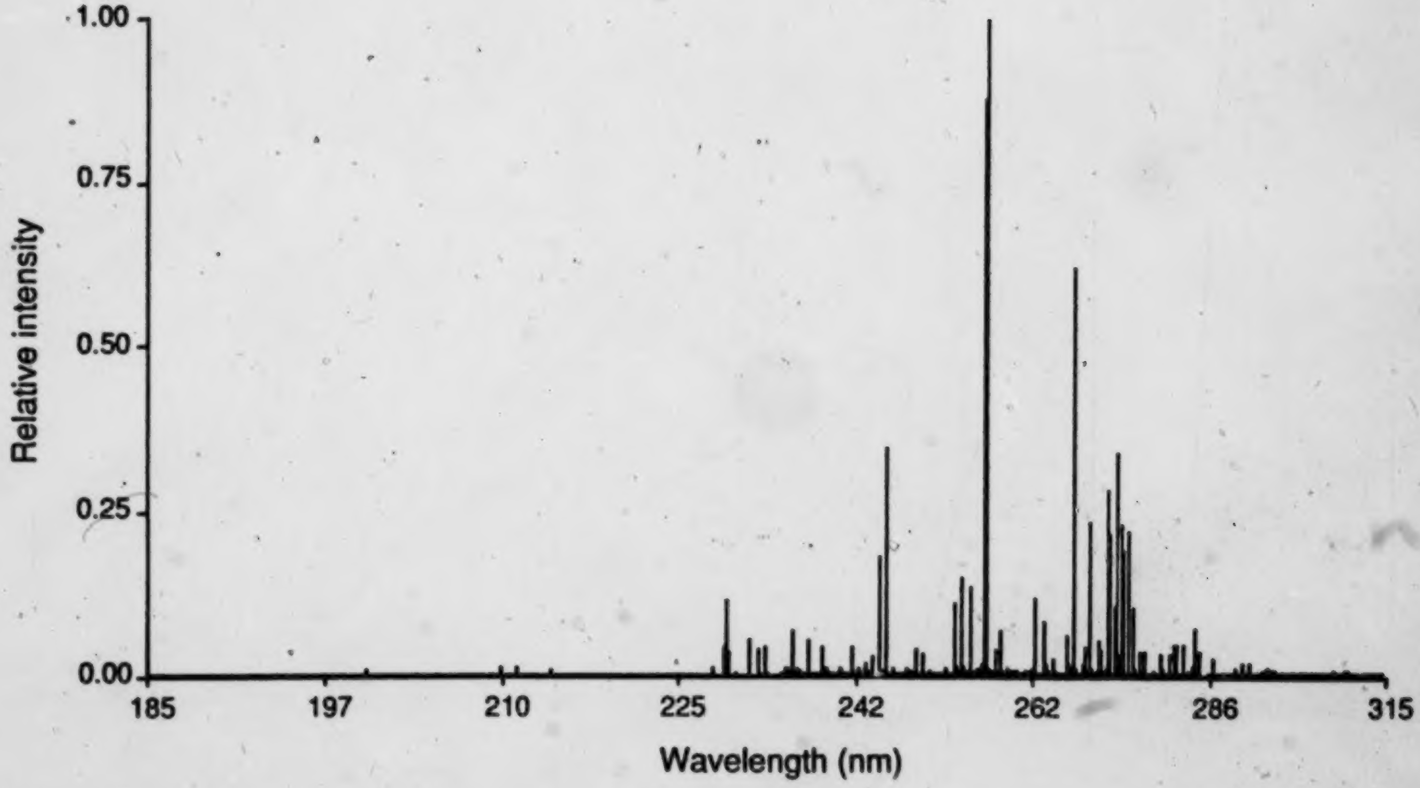


Figure F-22. The ICP ultraviolet spectrum of Zr.